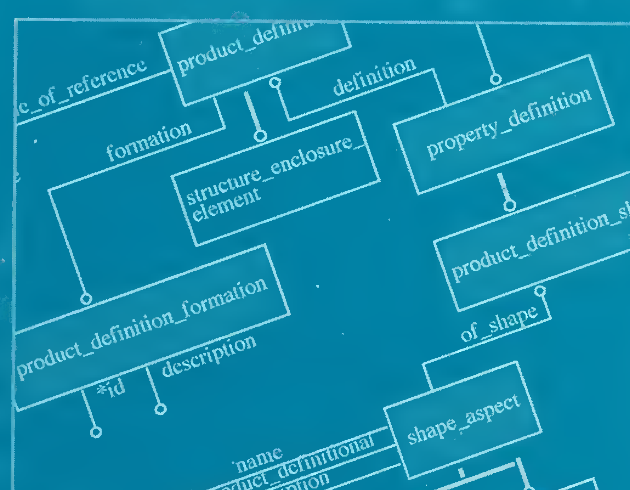
NIST SP 939
U.S. Department of Commerce • Technology Administration • National Institute of Standards and Technology

STEP
The Grand Experience



Edited by
Sharon J. Kemmerer

NIST Special Publication 939

STEP
The Grand Experience

Edited by
Sharon J. Kemmerer

Manufacturing Engineering Laboratory
National Institute of Standards and Technol
Gaithersburg, MD 20899-0001

July 1999

U.S. Department of Commerce
William M. Daley

Technology Administration
Gary R. Bachula, Under Secretary for Technology

National Institute of Standards and Technology
Raymond G. Kammer, Director

National Institute of Standards and Technology
Special Publication 939
Natl. Inst. Stand. Technol.
Spec. Publ. 939
187 pages (July 1999)
CODEN: NSPUE2

U.S. Government Printing Office
Washington: 1999

For sale by the Superintendent of Documents
U.S. Government Printing Office
Washington, DC 20402

DEDICATION

Because this Product Data Exchange effort has already crossed into its third decade, the United States standardization participants have felt the unfortunate loss of their colleagues. Many of these individuals were still active contributors to the national and international efforts when they died. This dedication serves as a small token of our appreciation for their professional contributions to the national and international product data exchange standardization efforts, and for their personal contributions in touching our lives.

Robert Colsher
(IGES)

William Conroy
(IPO Chair, NIPDE General Manager, US TAG Chair)

Bert Gibbons
(Electrical/Electronics)

Dick Justice
(US TAG Chair)

Bryan Martin
(Validation)

Richard Winfrey
(Presentation)

Timothy Wise
Liaison for TC172/SC1 and TC184/SC4/WG3/T9

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

This document is the result of a comprehensive effort by the National Institute of Standards and Technology (NIST) to document the development of STEP--Standard for the Exchange of Product model data. More than two dozen individuals inside and outside of NIST actively contributed to this document. Dr. Richard Jackson, Director of the Manufacturing Engineering Laboratory, initiated this effort, making the following remarks as an introduction to the task:

> "I think it's time for us to take stock of what has transpired with STEP. After more than a decade of technical and standards committee work in product data exchange, it is time for us to produce a definitive work on this topic. This work should include clear, concise, illuminating discussions of the technical issues, the solutions and the standards..." [10/1996]

Although the text will focus on the role of NIST in the STEP effort, it will also describe the role and efforts of the many types of partners that have worked with NIST to make STEP happen. In addition, this document will examine a possible path for NIST to take in determining its future involvement with STEP or other similar standards. NIST's effort in product data exchange standardization has helped to expand its role in physical measurements and calibrations into the arena of:

- Validation, conformance, and interoperability testing.
- Developing information technology (IT) tools that support the implementation of IT standards.
- Disseminating information about the standards.
- Developing a programmatic thrust in information technology metrology for manufacturing.

A further objective of this document is to inform the reader about the importance of standards in facilitating the ability of companies to manufacture and use their products. As the world moves into the twenty-first century, new manufacturing technologies are needed to improve productivity and competitiveness. In this information and computer age, companies exchange and share information across the country and the world. This capability is needed to manufacture complex products, such as automobiles, airplanes, ships, and buildings, which are produced today. To meet production deadlines, computer-aided design and manufacturing tools are used to move products from concept, through design, prototype, manufacture, testing, and support that is required by the customer. This information exchange process must be accelerated if it is to be useful. Today, existing products and technologies are often replaced before their useful life has expired. This is driven, in part, by the competitive nature of the manufacturing marketplace.

In this age of agile manufacturing, concurrent engineering, and teaming, the ability to share product data information quickly and efficiently among a variety of different computing environments is critical to any collaborative effort. Such collaboration needs to take into account efforts either within a company or across different companies cooperating in normal business and commerce. The representation of product data in digital form is a technology that is basic to both a company's internal plans for integration and its external relationships with the world. Product data exchange is the essential component to implement the standards and technologies required to make the collaboration applicable for manufacturing.

NIST has taken a lead position in developing STEP because of its historical mission to promote U.S. economic growth. NIST has done its best by working with industry to develop and apply technology, measurements, and standards. Specifically in the last few years, the NIST laboratories have increased their efforts to address the infrastructural needs of the information technology and manufacturing industries. STEP is an ideal example of a set of standards that integrates both industries. NIST recognizes that developing standards such as STEP must be accomplished in the international arena because of the ever-increasing worldwide economic dependencies.

From its start, the STEP project benefited from a large number of experts that brought with them a wide range of knowledge and skills. One commonality was the enthusiasm, dedication, and hard work that all put into the effort. When the committee started, few if any of the team had any significant experience in developing standards. In one sense, this was an impediment since much time was spent learning the intricacies of the ISO process. But in another sense it was a major benefit since the secretariat and the technical team was unfettered with how things were typically done in other standards committees. The STEP effort pioneered several accomplishments. STEP was the first ISO standard to:

- Use formal information modeling techniques in its development.
- Publish a standard for an information modeling language.
- Include digital information in its normative form.
- Include a specification for conformance testing.

This work traces the history of the development of STEP. Successes, setbacks, and outstanding issues are discussed. Ultimately, NIST believes STEP is succeeding, both as a technical solution to the problem of product data exchange and as a contribution to the standards development process.

1.2 DOCUMENT APPROACH

In Chapter 3 the reader will be introduced to a group of individuals from the PDES/STEP community, known as the Ad Hoc Complainers and Gripers Committee [1]. As part of their argument for a particular approach to integrate the many standard parts of STEP, they made an analogy. This same analogy aptly describes the approach taken for this document:

> "The development of PDES[1] can be likened to editing a book by several authors, for example a book resulting from a conference. The book could be put together by:
>
> 1. The editors merely collect all the presented papers into one volume.
> 2. The editors select a subset of the presented papers and publish these as one volume.
> 3. The editors decide on an outline of each chapter to be written, request the authors to write their chapter following the outline, and publish the resulting collection.
> 4. The editors decide on a theme for the book and prepare an outline for each chapter; commission authors to write their chapters following the outline; edit the chapters to remove inconsistencies between chapters, to provide adequate cross-reference between chapters, to fill in any missing ideas and to provide a consistent "style;" and finally publish the result."

To learn what scenario was recommended by the Complainers and Gripers Committee for STEP integration, the reader needs to wait until Chapter 3; however, for the purpose of compiling this document, the editor has attempted to pursue scenario 4. Appendix D, About the Authors, allows the reader to identify the contributors by chapter, to directly target any questions that may arise from the reading. *The Editor attempted to preserve as much of the authors' technical and historical opinions as possible; however, to move toward a work that integrates as scenario 4 suggests, some of the authors' opinions associated with a particular chapter may no longer reflect the authors' original intent.*

---

[1] PDES is an earlier acronym for the STEP standardization work carried out in the United States. Today, PDES stands for Product Data Exchange using STEP.

1.3 DOCUMENT CONTENT

In the past, companies conducted business mostly using their internal, proprietary formats. As we move into a global marketplace, we are beginning to see a dramatic paradigm shift toward open international standards. Figure 1-1 shows a view of these trends. Standards are no longer trailing technology and are no longer an after-the-fact documentation exercise. Standards provide a critical foundation in achieving effective and efficient communication within and among companies.

The processes of developing standards and the way we describe products have come a long way. An attempt to parallel the technology growth with the standardization is not a trivial exercise. You will read in later chapters how STEP development was, and still is, an experiment in parallel standardization with progressing technology.

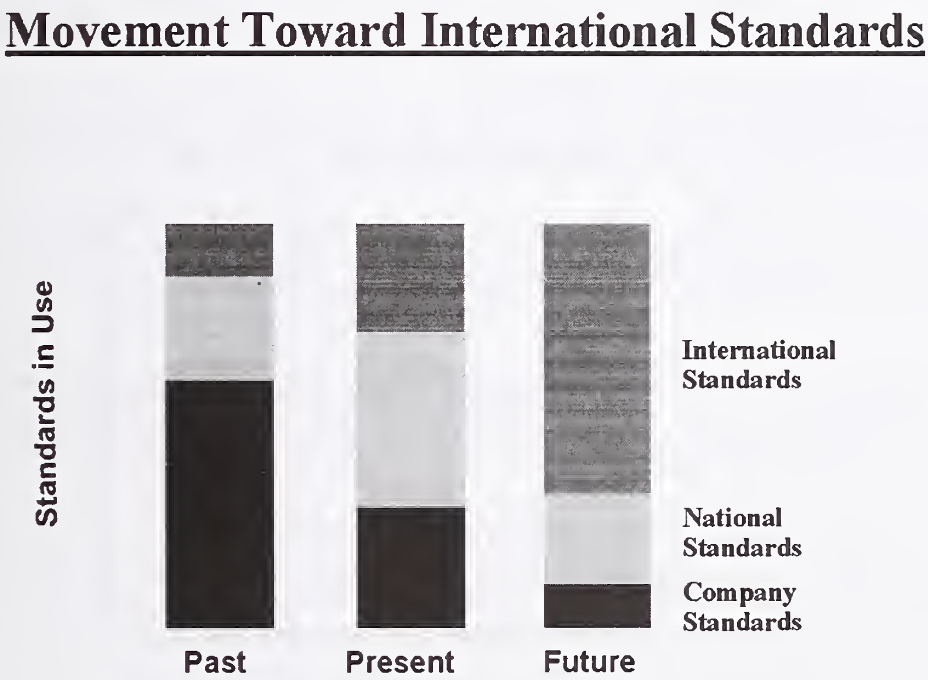


Figure 1-1: Movement toward International Standards (contributed by PDES, Inc.)

The potential impact of STEP is enormous and its effect is just beginning to be seen around the world. With more than 50 production implementations in the U.S. and Europe, STEP is already reducing lifecycle costs and product time to market, as well as providing increased flexibility and agility. A few examples include:

- Lockheed Martin's F-22 program has shown consistent savings using STEP: 50% process saving for composites, and projected 27% savings on tool design for CAD/CAM systems.
- Boeing, in its 767 and 777 programs, has shown a 75% time savings in processing designs from engine suppliers using STEP.
- Boeing, in its C-17 program, has reduced the time to transfer bill of material data from weeks to minutes using STEP [2].

NIST hopes this text will capture both the pain and the gain required to get where we are today in information-managed product data exchange.

The remainder of this document is broken into four primary sections:

- Chapters 2 and 3 provide the basis for understanding product data exchange standards and the developmental history of STEP.
- Chapters 4 through 8 focus on aspects believed to be technical innovations for developing STEP and perhaps for contributing to the promotion of information technology into sectors such as manufacturing, electronics, and process plants. These chapters also provide more technical depth specific to ISO 10303 [3].
- Chapter 9 reviews the international standards development organization (SDO) responsible for STEP and associated standards development tools.
- Chapter 10 presents a look to the future: what lies ahead as impacted by what went before. Chapter 11 reflects on the past and NIST's involvement in the making of STEP.

The document concludes with glossaries of acronyms and terms, additional references that offer background reading for some of the chapters, a brief biography on each author, an index, and a bibliography of chapter citations.

The following provides a brief overview of each chapter.

Chapter 2 covers the history of product data exchange standards that lead up to the initial investments in STEP.

Chapter 3 characterizes the phases of development that occurred during the process of reaching consensus on an international standard for product data exchange. The focus of development shifted a number of times before the architecture of STEP emerged. This chapter addresses the shifts, the technical implications, and the ultimate decisions that led to STEP as we know it today.

Chapter 4 provides an overview of the STEP architectural components and methodology and describes the requirements addressed by the STEP architecture and the governing principles of that architecture. Now that the Initial Release of STEP has been an international standard for several years, some of the perceived problems with the current architecture are also discussed.

Chapter 5 views modeling as crucial to the success of STEP; however, modeling is a very complex area and continually under study. The abundance of conflicting requirements and different proposals, each with different paths, helped to shape EXPRESS, the modeling language that was created for and used by STEP. Ultimately, SC4's[2] ideas came to encompass both compromise and innovation in the difficult area of modeling. Chapter 5 then elaborates on EXPRESS as one of the cornerstones of STEP. It considers how the SC4 community has benefited from developing EXPRESS and the limits to which EXPRESS can meet the needs of automatic code generation and model validation.

Chapter 6 focuses on sharing data and implementing the standard versus modeling the information. Particular emphasis is given to the Standard Data Access Interface (SDAI), ISO 10303-22, the data sharing interface standard of STEP.

Chapter 7 explains the purpose and principles of the application protocol (AP) methodology and gives background on the decision to add APs to the STEP architecture. This chapter also provides a summary of the mechanisms for planning and managing AP projects and some of the lessons learned from the AP methodology.

Chapter 8 introduces the methods, tools, and integration of standards-based products as achieved through testing. The two primary approaches for achieving systems integration are conformance testing and interoperability testing. This chapter discusses the purpose of testing and describes the relationship between conformance and interoperability testing in the context of STEP.

[2] ISO TC 184/SC4: Technical Committee 184 on Industrial automation and systems integration, Subcommittee 4 on Industrial data.

Chapter 9 covers the methods, manpower, materials, and tools that contribute to the standardization process of STEP within the international standardization community. One should approach this chapter with the understanding and appreciation that EVERYTHING surrounding the standardization of STEP is huge in magnitude and done with a respect for complexity. Because typical ISO standards are shorter in length and smaller in scope, and most ISO subcommittees meet with less frequency and with a smaller number of technical experts, SC4 has had to find innovative ways to handle its standardization process.

Chapter 10 presents a vision of the future of STEP -- how it will impact industry and government beyond the year 2000. It includes both development and implementation perspectives, as well as thoughts on future product direction. It also describes how STEP will interface with related groups and standards in the future.

Chapter 11 is simply the epilogue. It summarizes the magnitude of the STEP effort, and NIST's role in that effort.

1.4 AUDIENCE

This document assumes a basic understanding of the standards development process. The intended audience for this document is standards developers. These developers may or may not have a basic understanding of STEP or of product data exchange. Specifically, some of the audiences who may be considered prospective readers include:

- ISO TC 184/SC4 liaison organizations.
- ISO/IEC technical committee liaisons to SC4.
- ISO TC 184/SC4 technical experts and their managers.
- IPO technical experts and their managers.
- US PRO members.
- U.S. industry associations and other candidate liaisons to SC4.
- Information technology vendors.
- International CALS community.
- STEP Centers around the world.
- Other national government agencies interested in product data standardization.

This text may also be useful as a teaching aid to introduce product data concepts and standards at the college level.

1.5 DISCLAIMER FOR THIS DOCUMENT

Any mention of a product, company, or service in this document is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology. This document is richer from drawing upon specific examples and from other literary works, other organizations' participation, and other product data tools and services. The reader should assume all references are to enhance the material presented, and to put in context NIST's role in standardizing product data exchange.

CHAPTER 2

IN THE BEGINNING... THERE WAS PRODUCT DATA EXCHANGE

2.1 EVOLUTION OF PRODUCT INFORMATION SHARING

"Before the dawn of the industrial revolution, engineering work was defined by a physical model of a product to be reproduced. For example, a worker manufacturing a rifle barrel would ensure that the dimensions of the barrel corresponded to a model barrel by using calipers to transfer measurements from one to the other. This method reinforced the concept of workers manufacturing specific product types rather than generic components of larger products.

> In 1801, Gaspard Monge wrote 'La Geometrie Descriptive' as the first treatise on modern engineering drawings. This included the theory of projecting views of an object onto three planes and the addition of size specifications to the shape descriptions. With the mechanical drawing, an objective standard of performance for workmanship was possible and thus the model could be eliminated. The drawing enabled the practice of designing a product with interchangeable parts to be created. Operations could then be performed using contractors that could manufacture different pieces to be assembled. This capability led to the fragmentation of the manufacturing process that exists to this day.
>
> The mechanical drawing concept has lasted for almost 200 years. As described above, the manufacturing process for developing quality products was interwoven with the method for describing the products. The drawing became the output of the design process and the input into the manufacturing process. Drawings were converted into process plans, which were converted into programs or procedures for the manufacturing operations. Thus, every process has its own view of the product data. These dissimilar views have made it difficult to feed back knowledge about different processes to the designer. In today's industrial enterprises, the lifecycle processes for a product are no longer all performed by the same group of people. In fact, the processes are distributed through a network of factories.
>
> As we move into the twenty-first century, new manufacturing technologies are needed to improve productivity and competitiveness. In this information and computer age, companies exchange and share information across the country. This capability is needed for manufacturing the complex products such as automobiles, airplanes, ships, and buildings that are produced today. There is a special consideration for accelerating this information exchange process since the existing products and technologies are often replaced before their useful life has expired as manufacturers compete in the marketplace. To meet production deadlines, computer-aided design tools are used to move products from concept, through design, prototype, manufacture, test and, ultimately the support that is required by the customer [4]."

The representation of product data has evolved slowly over these same 200 years (see Figure 2-1). Before 1800, a tangible physical model of a product defined product descriptions. The invention of the engineering drawings in the early 1800s led to more precise product descriptions. This precision increased productivity sixfold over using a physical model to define a product.

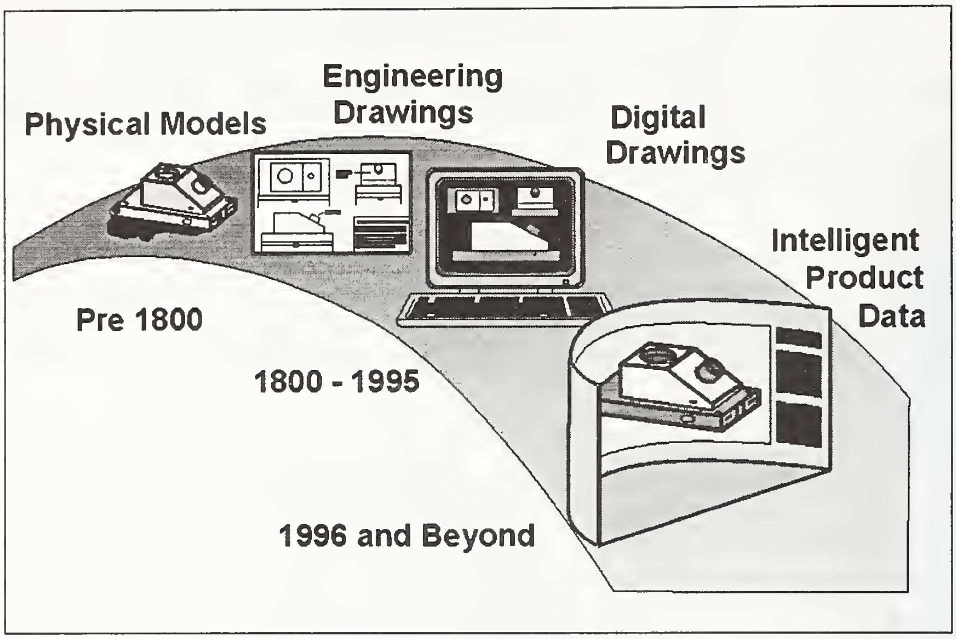
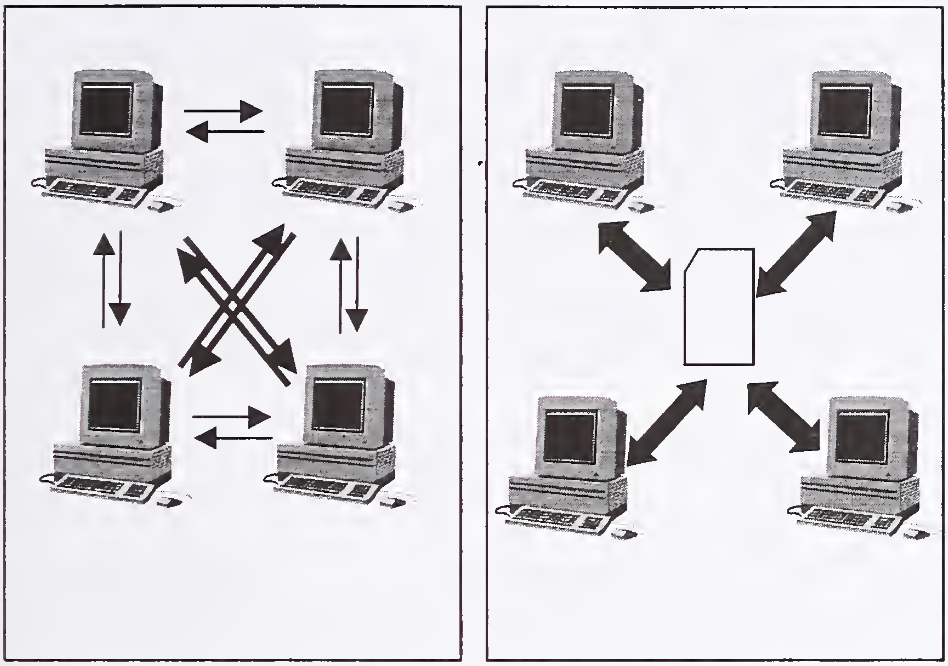


Figure 2-1: Evolution of Product Definition Capabilities

Drawings created with Computer-Aided Design (CAD) tools represented tremendous productivity gains over paper drawings, such as ease to revise and archive. CAD tools also opened new opportunities, such as enabling manufacturing instructions to be derived automatically and executed directly from the drawing. Nevertheless, as computer design and manufacturing tools proliferated to meet increasingly complex and diverse engineering needs, so did the formats each tool uses to capture and store product data. While paper drawings can be marked up by anyone with a pencil, a product model that can not be interpreted by the necessary CAD tool is useless. For organizations to share designs across various CAD and Computer-Aided Manufacturing (CAM) tools, they must be formatted in a manner that the tool can recognize. This requirement is becoming increasingly important in an age where large manufacturers often form joint ventures to address a business opportunity, and where partners in a supply chain are being called upon to deliver an increasingly complex array of services.

Most companies find it difficult to enforce the use of a common set of CAD/CAM tools within their organization, much less across (multiple) supply chains and among joint venture partners. Because of the lack of any common set of tools, a common format for neutral file exchange is needed. It is exactly this common format, as well as data access mechanisms that STEP hopes to provide. The cost benefits are suggested by the reduction in necessary translators shown in Figure 2-2. The figure illustrates that by using a neutral file exchange, the number of translators (for N systems) can be reduced from N^2 to N. Using a neutral standard for transferring information across systems drastically reduces the requirements for translators.

Efficiency of a Neutral Format for Data Exchange



... By Direct Translators **... By Neutral Format**

Source: Department of Trade and Industry, "Product Data Exchange, An Introductory Guide," Finallay Publications, UK.

Figure 2-2: Efficiency of a Neutral Format for Data Exchange

2.2 EARLY CONTRIBUTING EFFORTS

The quest for a common output format among design automation tools did not start with STEP. STEP in many ways can be seen as the culmination of various U.S. industry, government, and international efforts. For example, in the 1970s the X3/SPARC Committee of the American National Standards Institute (ANSI) contributed the notion that data should be described in a manner that was independent of particular uses or computer technologies. SPARC proposed a three-schema methodology within which one basic conceptual information model could be realized in a variety of computer technologies and presented to users through a variety of filters. These different views of the same information were called conceptual, internal, and external views. (See Figure 2-3 [5].)

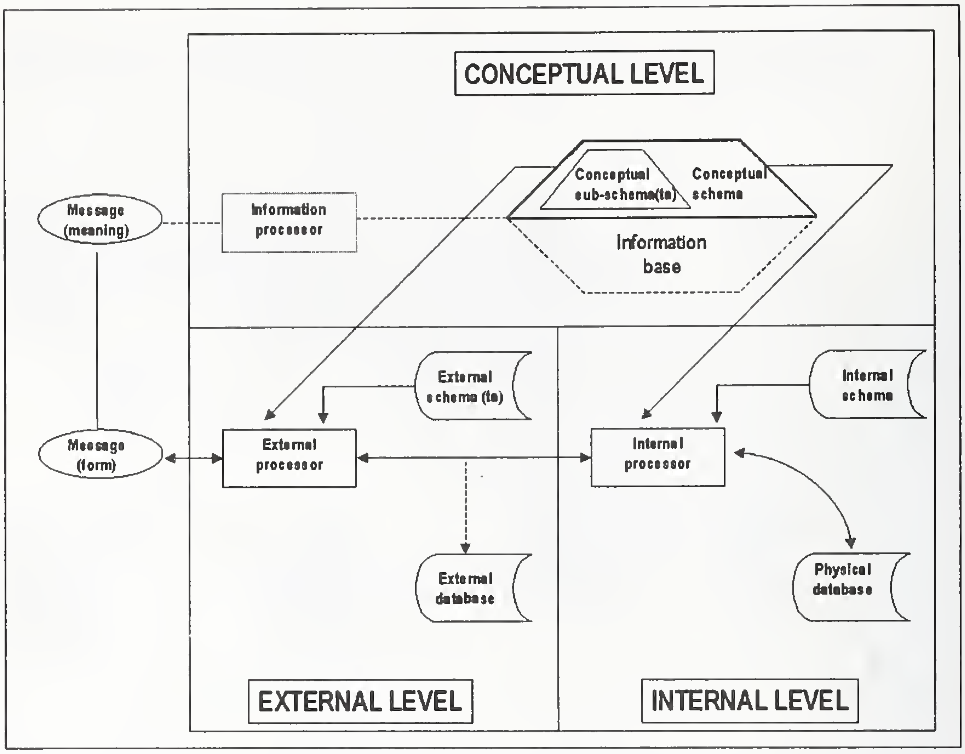


Figure 2-3: Three-schema Methodology

The U.S. Air Force built upon the ANSI/X3/SPARC methodology by developing formal methods for information modeling, as a part of its Integrated Computer Aided Manufacturing (ICAM) program. The intent of ICAM was to develop new manufacturing automation technologies that could lower the overall cost of procurements. The program determined that new systems engineering methodologies were needed for developing new technologies, which implied new methods of defining requirements. This work resulted in a suite of formal methodologies: IDEF0 for modeling activities, IDEF1 (later extended to IDEF1X) for modeling information, and IDEF2 for modeling system dynamics. ICAM awarded a series of contracts that required the use of these new systems engineering methodologies. Some of these contracts had direct impact on developing STEP. This held true for other programs as well. The Integrated Programs for Aerospace-Vehicle Design (IPAD) project funded by NASA, for example, had a geometry focus and is credited with being the first to make use of information modeling for systems integration.

ICAM and its subsequent contracts, including the Product Definition Data Interface (PDDI) and Geometric Modeling Application (GMAP) programs, contributed much to the tools and methodologies later applied to STEP. Other efforts contributed to the formal description of the information needed to be shared among CAD systems. The Computer-Aided Manufacturing - International (CAM-I) organization, through its Geometric Modeling Project begun in the early 1970s, contributed significantly to the formal description of Boundary Representation (B-REP) data. The result of the CAM-I funded work, which was a mathematical representation of standard geometry and topology, was considered ahead of its time and clearly captured more information than the typical CAD systems of the day could interpret. It was submitted to ANSI committee Y14.26[3] for standardization as a data exchange mechanism. The CAM-I specification did not contain an exchange mechanism, but a foundational description of the data that could be exchanged.

[3] The ANSI Y14.26 committee name was Digital Representation for Communication of Product Definition Data.

2.2.1 The Birth of IGES

In 1979, events took place that catalyzed the CAD vendor and user community to create the first national standard for CAD data exchange. CAD systems were less than ten years old, and there were only a handful of products with any significant market penetration. Even at this early stage, users were overwhelmed by the inability to share data among these tools and with their own internally-developed databases. In September 1979, frustration came to a head at the two-day Air Force ICAM Industry Days Meeting [6]. On the first day, a representative from General Electric (GE) challenged a panel of CAD vendors, which included ComputerVision, Applicon, and Gerber, in essence, to stop blocking progress and work together to enable an exchange mechanism. While this need was intuitive from a user's perspective, this was a very threatening proposition to the CAD vendors—who feared that sharing the structure of their databases publicly would be tantamount to giving away their competitive advantage. It would have been easy to gloss over the challenge; after all, the major vendors all had at least token representation on the ANSI committee responsible for CAD standards. Instead, the ComputerVision representative responded with a challenge of his own: If Boeing and General Electric (and perhaps others) would contribute the CAD translators they had already developed, the vendors would share their database structures.

What led to this offer was just the right mix of business motivation and hidden agendas. It just so happened that the evening before the CAD panel, a CAD vendor representative was busily recruiting employees for his (unannounced) new robotics company. In forming this company, he gained the user's perspective: his product was going to need to have access to CAD data! If he could set the wheels in motion for the CAD vendors to make their database structures public, his new company would have a better chance at success; however, an exchange standard was also in the CAD vendors' best interest. The CAD vendors tried to differentiate themselves based on loyalty to their customers; this also had the negative effect of dividing the end users into camps. There were large Navy contracts looming on the horizon, and no vendor wanted to look unresponsive to customer requirements.

In the evening after the panel, several interested parties gathered in a smoke-filled room and asked themselves if a common translator was really possible. The room had the right mix of people and ideas at the right time. This included an Air Force ICAM, Navy, and NASA representative, each willing to fund $25,000 for such an effort. A National Bureau of Standards (NBS)[4] representative who, after a call to his boss at home for a sleepy approval, was willing to champion it as chair and coordinator. The IGES Organization was formed by NBS in the spring of 1980. With the fundamentals to a common translator decided, conversation turned to a name for this new translation project. The group nixed the suggestion "Universal Translator" to avoid offending those within ANSI who might have interpreted the project as a way to displace the years of effort already put towards a Y14 standard. A minimalist approach was suggested:

I - Interim, to suggest that it would not replace ANSI's work
G- Graphics, not geometry, to acknowledge that academics may come up with superior mathematical descriptions
E- Exchange, to suggest that it would *not* dictate how vendors must implement their internal databases
S- Specification, not to be as imposing as a standard

The panel reported on the second day, and the wheels were set in motion to create an "IGES." Once the panel admitted that a common translation mechanism was possible, it was impossible to stop the momentum of the customers' enthusiasm and expectations. Applicon and ComputerVision agreed to open up their internal databases, GE offered its internal database structure, and Boeing supplied the structure of its Computer Integrated Information Network (CIIN) database. Both GE and Boeing contributed their existing translators. A core team was formed which included representatives from NBS (Roger Nagel), Boeing (Walter Braithwaite), and GE (Phil Kennicott). Team members had worked closely with each of the vendors on internal integration projects. This prior experience built the expertise and trust needed to craft a solution in a very short time, and neither vendor felt it gave an unfair advantage to the other.

---

[4] Department of Commerce's National Bureau of Standards was renamed the National Institute of Standards and Technology (NIST) by the Omnibus Trade and Competitiveness Act of 1988.

Soon after the ICAM Industry Days, NBS called an open meeting at the National Academy of Sciences (October 10, 1979). Around 200 people attended to herald the birth of IGES. There was an atmosphere of extraordinary excitement, although not everyone was supportive. In addition, although it was hotly debated, the name was accepted eventually with the minor change from "Interim" to "Initial."

After two critical reviews, the IGES team released their first draft in 1980, containing geometry, graphical data, and annotations. The IGES specification was brought to the ANSI Y14.26 committee for standardization, an action which forced the committee to try to reconcile the very different views embodied by the IGES work and the CAM-I boundary-representation description effort. When the first version of IGES was adopted as a standard (Y14.26M-1981[5]), it approved the IGES draft with the CAM-I work attached (but not integrated) as the fifth section, entitled "Section five - Basic shape description." Subsequent versions of ANSI Y14.26M omitted this fifth section .

Although it had funded the work, CAM-I recognized that Section five of IGES Version 1 was not compatible with the shape description methods used in current CAD systems. CAM-I therefore developed an alternative specification, resulting in the Experimental Boundary File (XBF) of 1982 [7]. This specification used the same format and file structure as IGES, but allowed for the exchange of solid models many years before IGES itself acquired that capability. The CAM-I XBF influenced various later efforts in solid model data exchange, and ultimately affected the STEP part, ISO 10303-42 [8].

Brad Smith recalls... When I think back on the early versions of IGES, I remember one of those editing sessions. It was Version 3.0. We were running late as usual, and I had committed to be in London for an SC4 meeting. I had also just taken delivery of the first laptop computer we ever had in our group. I decided to take the master copy of the IGES document with me, along with the laptop, so as to have the editing done by the time I got back home. At the last minute, I realized the laptop only ran on 115 VAC power, so I hit an electronics store on the way to the airport. The first night in the London hotel room, I felt rather smug as I settled into the IGES editing after hooking up all the adapters, converters, and cables. Unfortunately, I soon noticed the plastic case on the new power converter had started to melt. Not wanting to stop, I put the converter into the small room refrigerator, slammed the door on the cords, and went on editing the document for the next three days.

Work in CAM-I had started even earlier on the Application Interface Specification (AIS), a proposal for a standardized programming interface to CAD modeling systems. This proposal eventually spent the period 1992-1994 as an ANSI Draft Standard for Trial Use [9]. The AIS has recently been released to the Parametrics Group in ISO TC184/SC4/WG12 for extension and updating as part of the new parametric capability they are developing for STEP (see Chapter 10).

Once the technical content of any standards document has been agreed upon, most people feel the job is done. Few realize how much work goes into the final editing of the document. It is an exacting task requiring attention to a multitude of small details. The sheer size of the IGES standard, for example, with its many figures and internal references made the job of editing quite a nightmare. The product data community owes an enormous debt to people like Bob Colsher, Joan Wellington, JC Kelly, Phil Kennicott, Dennette Harrod, Brad Smith, Gaylen Rinaudot, Kent Reed, and others who dedicated themselves to final production of each edition of IGES. Each spent days, weeks, and months of unreimbursed personal time laboriously editing and re-editing those chapters of text and figures.

2.2.2 Product Definition Data Interface (PDDI)

IGES provided a practical first solution for CAD data exchange, complete with an exchange file format. The speed with which this first draft was developed was remarkable! It may have been due, in part, to the relatively limited scope of the specification and the small size of the committee developing it. An additional contributor was the contract requirement to publish a document within three months of the contract award. Once it fell under the scrutiny of an ever-broadening community, weaknesses were identified that eventually justified embarking on a new standard, which could break tradition with IGES. The Air Force ICAM program again made a significant

[5] Since revised and republished as ANSI/US PRO/IPO 100.

contribution to the evolution of product data exchange standards, this time through its Product Definition Data Interface (PDDI) contract with McDonnell Aircraft Company. The purpose of PDDI was to develop a replacement for *blueprints* as a communication mechanism between engineering and manufacturing. It sought to replace all information found on a blueprint (more commonly known as an engineering drawing today). PDDI developed a set of information models, a modeling language which contributed to EXPRESS [10], an exchange file format that separated that data being exchanged from its definition, and a mechanism for applications to share data.

One of the tasks of this contract involved an evaluation of IGES in the context of its current implementations. This resulted in a thorough report [11] and numerous constructive requests for changes to IGES. The evaluation activity helped the community clearly define IGES's shortcomings:

- Flavorings - IGES contained several ways to capture the same information, which made proper interpretation largely dependent on the particular "flavor" of the pre- and post-processors.
- File size/Processing time - IGES was criticized heavily for requiring large files that took hours or even days to parse with the average computing power available at the time.
- Loss of information during exchange - Information will inevitably be lost when information is passed between two CAD systems with inherently different capabilities.
- Lack of discipline, architecture - There was a perception that IGES was developed without rigorous technical discipline, and that formal information modeling would be useful.
- Upward compatibility - The need for generations of processors to parse files compliant with earlier versions of IGES thwarted the breadth and rate of change in succeeding versions.
- Automated a paper system - IGES was seen as a method to exchange engineering drawings, but not capable of capturing complete product data (including administrative information) to enable more sophisticated automation which would reduce or eliminate human intervention to translate.

Although PDDI was a research effort from 1981-1987, it contributed understanding, mechanisms, and models to what later evolved into STEP. It served as the "kick in the pants" for the IGES Organization to think "What's next?" Those from the PDDI team had the opportunity to make a real impact on future PDE standards.

Additional shortcomings in IGES were later identified in a paper by Peter Wilson:

- Subsetting - Vendors selected and implemented only portions of the whole of IGES, thus making exchange between two systems impossible without prior agreement on what was to be exchanged.
- Processor testing - There was no mechanism for testing processors or resolving errors between two processors [12].

There was a real, long-term problem with IGES that would be difficult to fix: IGES communicated the lines and symbols appearing on an engineering drawing (except for some electrical concepts such as connectivity), but it failed to communicate the meaning of the information the engineering drawing was created to convey. The PDDI study revealed that *product features* must be transmitted with the geometry so that computer-based applications could "understand" the engineering drawing. For example, an application looking at an IGES representation would see merely a circle on a part. The desired result was to be able to distinguish whether that circle was a surface mark or a hole.

2.2.3 Subsetting and Application Protocols

The use of formalized subsets of IGES entities offered one approach to improving the quality and predictability of translations. NBS, under sponsorship from U.S. Department of Defense Computer-Aided Acquisition and Lifecycle Support (CALS) Program, led the IGES Organization[6] in building IGES subsets and applications for Defense. The

[6] The work on IGES Version 1.0 required the creation of two committees – a Working Committee to extend the capabilities of IGES and repair errors uncovered in that original version and a Steering Committee to oversee the

U.S. Department of Defense eventually stipulated IGES subsets for various application areas, such as technical illustrations and electrical/electronic applications, within their CALS suite of military standards. Subsets allowed IGES processors to be classified by the functionality that they could support *entirely*, and acted as a predefined written agreement between a sending and receiving party.

STEP's concept of application protocols (APs) grew from the lessons learned regarding IGES entity subsets and early IGES Architecture, Engineering, and Construction (AEC) application protocol work done by NIST for the U.S. Navy. Chapter 7 is devoted in more detail to the concepts and benefits afforded by STEP application protocols.

2.3 OTHER INTERNATIONAL PLAYERS

Several international efforts also contributed significantly to the evolution of product data exchange standards.

2.3.1 AECMA Report of Geometry Data Exchange Study Group

In 1977, the European aerospace industry recognized a major problem in exchanging shape representation on collaborative projects. The European Association of Aerospace Industries (AECMA) developed a common exchange format that allowed the collaborating companies to exchange simple surface geometry. The format was used on a few occasions, but the advent of more complex surface types, integrated into vendor systems, caused it to fall into disuse. Even so, there was good work done by AECMA. The United Kingdom contributed the AECMA Report of the Geometry Data Exchange Study Group to the International Organization for Standardization (ISO) effort for building an international product model data standard [13].

2.3.2 Flachenschnittstelle des Verbandes der deutschen Automobilindustrie (VDA-FS, VDA-IS)

The Germans standardized Flachenschnittstelle des Verbandes der deutschen Automobilindustrie (VDA-FS), which addressed the exchange of free form surfaces and free form curves needed by the automotive industry. VDA-FS was based on IGES but offered a competing exchange file format to that of IGES. The VDA was created in 1982 to increase the efficiency of the design process and usefulness of CAD/CAM systems. The Germans brought VDA-FS to the international table to contribute toward the international product model data standardization effort [14].

The German automotive industry, through VDA-IS (IS-IGES Subset), defined subsets of annotation entities that were relevant for various applications in automobile manufacturing. These subsets were created so that compliance could be tested. The particular data exchange requirements met by these subsets included: drawing information, two- and three-dimensional geometry, and analytic and free-form surfaces [15].

2.3.3 Standard d'Echange et de Transfert (SET)

The French Standard d'Echange et de Transfert (SET) project started at Aerospatiale in 1983. Aerospatiale needed a common database capability across its different CAD systems. They did a formal test of IGES and found it did not work. To be a little more precise, they tested the first beta IGES implementations from two vendors which, according to documentation, had implemented only points, lines, arcs, and text notes. (A major amount of information on an engineering drawing would of course be lost even if these few entities had been implemented completely and correctly!) From this test, Aerospatiale concluded that it was the *IGES specification* that did not work. The result was a French effort to write a specification, standardize it, implement it, test it, and support its use in production. Designed to address the difficulties using IGES, the primary industrial drivers of SET were automotive and aerospace industries. The standard represents the results of the requirement to exchange data

---

operation. The Working Committee had two major subcommittees: an Edit Committee and a Test, Evaluate, and Support Committee. Together the Working and Steering Committees were referred to as the IGES Organization.

between different CAD/CAM systems, and from the need to archive these data. Version 1.1 of SET was put on the international table to contribute toward the international product model standard. It contained:

- Detailed specifications for the mechanical area.
- Supplementary information about the data structures and concepts employed.
- Provided rules and recommendations concerning specifications to ensure coherence in future developments [16].

Association GOSET is an organization established by industry and government in France to support continued development, maintenance, and implementation testing of SET. GOSET representatives are also active contributors to developing STEP and testing services to conformance test ISO 10303 [17].

2.3.4 CAD*I

In 1984, the European Commission funded an ESPRIT project called CAD Interfaces (CAD*I), with twelve participating organizations from six European countries. The project worked mainly in product model data exchange and on data exchange for finite element analysis. As in STEP, the transfer of data was based on the use of schemas defined formally using a data modeling language. In 1987, this project achieved the first ever transfer of boundary-representation solid models between different CAD systems. CAD*I participants were involved in the development of STEP from the beginning of the work of ISO TC184/SC4, and some of them are still active today. Much of the shape modeling capability of STEP is based on CAD*I work [18], and the project also had a significant influence on STEP developments in the finite element area.

2.3.5 Why Not Adopt IGES Worldwide?

The following realities became drivers for a common international standard:

- Global commerce and increased outsourcing making data exchange more critical.
- More complex products which require coordinating among multiple engineering disciplines.
- Multi-use software, e.g., design or engineering systems that apply to multiple industries and applications.
- Reliance on suppliers at all phases of product development.
- Need for lifecycle support.

Moreover, these are the generalities. As cited earlier, IGES as a world contender for international standardization, also had many technical flaws.

2.4 THE PDES INITIATION EFFORT

By 1984, many of these efforts had produced enough results to be compared, and an international community was preparing to form a committee in hopes of creating a common solution to CAD data exchange. In May of 1984, a late night meeting of the IGES Organization Edit Committee was held. The outcome: Kal Brauner, the Boeing representative was tasked to write a paper on what the next generation of IGES might look

Larry O'Connell (then of Sandia Laboratories) recalls... I remember the IGES quarterly meeting aboard the landlocked Queen Mary liner near Long Beach, CA. Most of us slept in staterooms aboard the elegant vessel and strolled to the daily plenary session after power walking around the ship and having breakfast on board. Much of the paneling in the less pricey staterooms was bird's eye maple. At least one of the plenary sessions was held in the grand ballroom under massive crystal chandeliers. Many representatives from across the Atlantic and some from across the Pacific attended to give the conference a truly international flavor. Brad Smith outlined his notion of what should be done to expand the scope and vision of the next version of "the standard." In the months that followed, a select few (guided by Kal Brauner of Boeing) began defining the requirements of what is now known as ISO 10303. In early 1985, the Queen Mary was a fitting setting for the launch of such a gigantic venture.

like without the IGES constraint to provide upward compatibility for processors. This informal request was in response to pressure from PDDI results and European efforts. The first Product Data Exchange Specification (PDES) report was issued in July of 1984, and was followed by a second report in November of 1984. These reports laid the groundwork for the PDES Initiation Effort, which, similar to PDDI, was considered a theoretical exercise at building a standard based on a broader automation goal and the discipline of information modeling. The PDES Initiation Effort used a simple machined part as its focus for both the "logical" information being captured and the "physical" mechanisms of data exchange. It also included an Electrical Schematic Application model. The Initiation task validated, through modeling, the concept that electrical connectivity and mechanical "joining" both shared a common topological model structure.

Those individuals involved originally assumed that this next-generation standard would be IGES Version 3. Instead, the work spawned a separate U.S. national effort known as PDES. PDES was eventually the specification for the international effort led by ISO TC184/SC4 responsible for developing and standardizing STEP. Chapter 3 provides more detail on PDES impact and this initiation effort.

2.5 HOW DID ELECTRICAL CONTENT FIND ITS WAY INTO STEP?

One might wonder how electrical content ended up in an ISO, rather than an International Electrotechnical Commission (IEC) standard. As with STEP, the roots trace back to IGES. The original vision for IGES included easy access to all machine-readable product data from any CAD tool, including data about electrical and electronic products. It was not until the second version of IGES that a very preliminary attempt was made to accommodate electrical connectivity information. Under the leadership of the IGES Organization Electrical Applications Subcommittee[7], developers began using information modeling in 1983 to improve the quality of electrical constructs in later versions of IGES.

Aside from the quality of the constructs, the EAC was concerned about overlap and duplication with other standardization efforts. In late 1983, the EAC met with the Institute for Interconnecting and Packaging Electronic Circuits (IPC) in an attempt to coordinate efforts. It was decided then that IPC would continue to focus on the CAD-to-CAM interface and the IGES EAC would focus on the modeling and CAD to CAD issues. Members of the EAC also heard of attempts by the silicon foundry to develop an interchange format for integrated circuit designs, and many wondered if that effort would duplicate, complement, or conflict with what was being developed for IGES.

2.5.1 Harmonization Activities

In April of 1984, The Institute of Electrical and Electronics Engineers (IEEE) Standards Coordinating Committee called a meeting that further drew the IGES Organization into a dialog with other standards efforts. Of particular interest was closer coordination between IGES and the relatively new Electronic Design Information Format (EDIF) effort. The EDIF representative at this meeting declined an offer of joint participation, for fear that standardization activities might delay the EDIF development schedule—a factor that has continued to impede, from both the IGES and EDIF sides, true coordination among related standards efforts. Other electrical standards represented included the IEEE Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) authorized by IEEE Project Authorization Request (PAR) P1076, the Abbreviated Test Language for All Systems (ATLAS), and the Tester Independent Support Software System (TISSS).

At about the same time, a representative from Westinghouse began reaching out to other related standardization efforts across the Atlantic Ocean, and authored several related papers that were published by CAM-I. He developed contacts that led to discussions between the IGES Organization EAC and the IEC TC3, Documentation and Graphical Symbols. In particular, NBS along with other IGES officers attended a meeting of TC3 subcommittee

[7] Later known as the Electrical Applications Committee (EAC).

SC3B in Los Angeles. This later facilitated the involvement of TC3 in the ISO/IEC Joint Working Group within ISO TC184/SC4.

Many organizations, including ANSI, and numerous individuals tried to find ways to increase the awareness and cooperation among related electrical standardization efforts with little measurable success. Each group working on some aspect(s) of the standardization for electrical and electronic product data had a set of volunteers, their sponsors, and a clientele to whom they felt they owed their scheduled deliverables. For the most part, no two efforts were initiated with the same goal, but rather were extended into overlapping territory in response to the needs of their users. Furthermore, some of the sanctioning standards bodies depended in some part for revenue from the sale of standards documents. A certain amount of jealousy about a perceived hierarchy of organizations also hampered some of the willingness of people at the working level to share results and efforts. The resulting array of conflicting and overlapping standards prevented the market from supporting any cohesive standard interchange methodology, and left much of the burden of data exchange on the shoulders of manufacturers who used electrical CAD systems.

In February of 1988, ANSI/ASME Y14.26 (the same committee that standardized IGES) raised the concern to ANSI management in a letter which stated:

> "...we are concerned that there are concurrent overlapping standards activities that are not coordinated. Of particular concern are the Initial Graphics Exchange Specification (IGES) Electrical Application subset, the Electronic Design Interchange Format (EDIF), the Institute for Interconnecting and Packaging Electronic Circuits (IPC) 35X series of specifications and the VHSIC Hardware Description Language (VHDL)..."
>
> While the standards cited were not the only efforts of concern, they were specifications which ANSI itself had authorized and which the government called out in CALS military standards. This letter led to a "Harmonization" meeting at EIA Headquarters in May, 1988. CAM-I's Electronics Automation Program (CAM-I EAP) Manager followed by offering to champion the effort. Participants included Boeing, McDonnell Douglas, Allied Signal, Eastman Kodak, Hewlett-Packard, Northrop, The Plessey Company, Westinghouse, and NIST. In February of 1989, the EIA issued results of an Evaluation Report entitled "Harmonizing CALS Product Data Description Standards."

The CALS/EIA Report found "...far more overlap than...anticipated... EDIF overlaps one or more other standards 78 times." The Report offered a matrix showing which lifecycle steps were captured by which of the four ANSI standards, carved out a scope for each standard based on this matrix, and declared harmonization effectively accomplished. This proposed solution was rejected by industry, as noted in CAM-I EAP R-90-EAP-01 which criticized the Report's conclusions. Milton Piatok of Boeing summed up industry's viewpoint in a letter to ANSI in 1989:

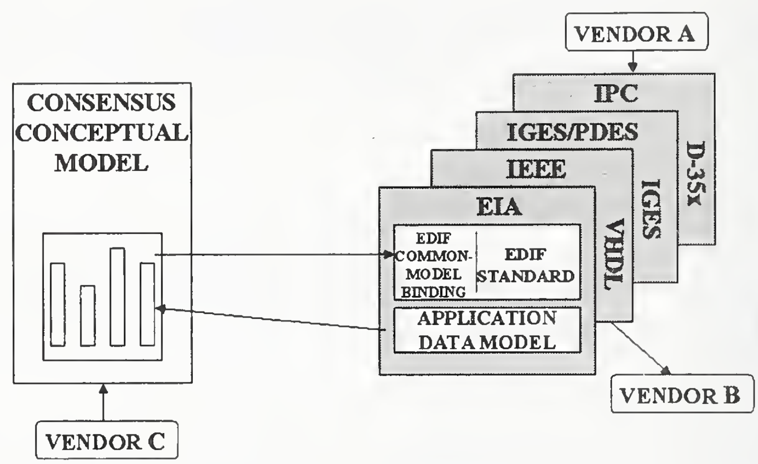
> "An electronics company which performs all the steps in the design process...using heterogeneous computer systems, work stations, and factory NC [numerical control] machinery and robots would have to support all four standards... At worst, this could mean not only having to implement the software to support each standard, but also having translators between each pair. ...Such an approach (if it were feasible) would be cumbersome, error-prone, time-consuming, and costly."

Howard Bloom recalls... When I was asked by the CALS program manager to lead the harmonization effort, I had no realization of the sensitivity of each of the standards development organizations. I had to be extremely careful at the early meetings with the words that I used. One phrase that might favor one specific standard sent the consensus building activity "two steps back." It took hard work, keen listening and great diplomacy to drive the activity towards accepting HPS-100. I don't think it would have been possible if I had been from any other organization other than NIST.

In November 1989, NIST accepted the leadership of the Harmonization effort, which was later formalized as the Harmonization of Product Data Standards (HPS) organization under the Industrial Automation Planning Panel

(IAPP) of ANSI. The HPS established three councils, to which NIST continued to serve as the Secretariat: Business Needs and Planning, Standards Development and Coordination, and Tools and Technology. Barbara Goldstein from McDonnell Douglas (now from NIST), led the Tools and Technology Council.

The major accomplishments of the HPS organization were to propose a methodology and a process for harmonizing the four ANSI standards, and to publish the first version of a coordinated information model as ANSI/HPS-100 "HPS Information Federated Model Descriptions."



"We do not expect to harmonize the formats which the EE standards have defined; but we do expect to harmonize the product information represented by these formats." Standards Development Coordinating Council, 7/91

Figure 2-4: The Operative Means to Harmonization

The HPS proposed the following process to guide harmonization, which reflects the group's early belief that the four standards would eventually be completely represented within STEP:

Process	Guidance for Harmonization
Gather Models	Gather verified conceptual models for the subject area of current focus from each of the relevant standards organizations.
Federate	Every element is added to federated model in data dictionary. Elements are classified. Unique, identical, and conflicting coverage is identified. Conflicts are resolved by creating generic elements that each conflicting element can be mapped to. Federated model contains each conflicting element as well as resolving elements.
Test	Define mapping between standards through the generic portion of the federated model. Create test vehicles (test cases) for the subject area of interest in the original standards. Run test: Sending Standard Format ↔ Federated Model ↔ Generic Portion of Federated Model ↔ Federated Model ↔ Receiving Standard Format Compare before & after files of test vehicles document mappings.
Harmonize	Derive harmonized model from tested, generic portion of federated model.
Submit for Standardization	Submit portions of harmonized model as candidate application reference models (ARMs) in STEP as they are ready. The harmonized model may also be submitted for national standardization. Hold public review.
Integrate with STEP	The portions of the harmonized model submitted for standardization within STEP will

Process	Guidance for Harmonization
	be integrated with STEP resource models in accordance with STEP procedure.
Develop APs, CDIMs	Develop application protocol (AP) and context-driven information model (CDIM) for subject area of interest. The AP will reference the mappings between the harmonized model and each standard. Identify information voids that none of the standards cover.

Table 2-1: The Harmonization Process (V1.1)

Both the information model and the guidelines for harmonization, later referred to as the "federation" to reflect the individual organizations' priority of autonomy, aided the groundwork for continuing international collaboration. The HPS was moved under the CIM Standards Board of ANSI and then deactivated as leadership in the area was transferred to the international arena under IEC Technical Committee (TC) 93. Through its working groups, IEC TC93 continues to develop a federated model to aid the interoperability of electrical information exchange standards. NIST representatives continue to play an active leadership role within IEC TC93 to build supporting electrical and electronic standards.

2.5.2 IGES Electrical Transition to STEP

To help interested manufacturers prepare their people for Computer Integrated Manufacturing using STEP, the EAC released the Layered Electrical Product Application Protocol (LEP AP), which was referenced in IGES Version 5.3. Initial EAC leadership to accomplish this release was from the Department of Energy Sandia Laboratories, and later from NIST by Curt Parks. The model resulted from a decade of development by scores of volunteers working under the IGES banner, plus many more working under various other banners. People working for and with the ANSI HPS mentioned above provided significant contributions to the model. A notable monetary and morale boost for this model came from the U.S. Naval Command, Control & Ocean Surveillance Center, Research Development Test & Evaluation Division which contracted for developing the IGES Hybrid Microcircuit Application Protocol [19]. This IGES AP was the most immediate predecessor of the LEP STEP AP.

2.6 LEGACY TO STEP

Even before the ANSI/HPS-100 model emerged, the early efforts of the IGES EAC provided some valuable general lessons learned about information modeling in a standard's setting:

- Team diversity - Need varied backgrounds (programmers, DBMS, subject matter experts, suppliers, customers, tester(s), a facilitator, scribes, visionaries).
- Committee resources - Need stable work force (of 6 to 12 people) having committed resources.
- Trained team. Need trained participants; otherwise, frequent methodology training (for newcomers) slows development.
- Strong leadership - Need knowledgeable oversight.
- Long-term commitment - Need long term commitment from management.
- Active communication - Need frequent and timely communication within the team.
- Strong public relations - Need public awareness of intent, schedule, and scope of effort.
- Necessary modeling - Information Modeling is not easy, but it seems necessary, though not sufficient, for Computer Integrated Manufacturing.

Other lessons contributed by IGES development…

- Standards can be produced quickly when the climate is ripe. What makes a "ripe" climate?
  - the standard has a very limited scope
  - only including in the standard what can be proven implementable (across at least three system implementations)
  - the playing field for consensus-building is relatively small – only a few CAD vendors in existence and only a few users applying CAD technologies
  - a high level of dedicated buy-in exists.

The technical legacy from IGES alone was plentiful for the next generation of product data standards:

- Requirements must be documented.
- Subsets were inadequate and application protocols were needed.
- Precision of the specification needed to be increased to reduce or eliminate interpretation needed to implement the standard.
- A product data exchange standard needed to have enhanced functional capabilities which included the need for:
  - Context and viewpoint (presentation does not equal meaning)
  - Specific semantics.
  - Three-dimensional (3D) solid model exchange
  - 3D tolerancing and dimensioning.
- Compliance assessment to the standard is necessary and requirements for it need to be present in the first edition of the standard.
- A migration path for legacy systems is a must.

2.7 CONCLUSION

NIST was a heavy player during this period of time. NIST employees held leadership positions in several of the supporting organizations described in this chapter:

- Chair of ANSI ASME Y14.26
- Chair of the IGES Organization
- Secretary of the IGES Organization
- Manager of Change Control to the IGES specification
- Co-Chairs of the Architecture, Engineering, & Construction (AEC) Subcommittee in the IGES Organization
- Secretary of the Architecture, Engineering, & Construction (AEC) Subcommittee in the IGES Organization
- Chair of the IGES/PDES Organization
- Secretary of the IGES/PDES Organization
- Chair of the IGES/PDES Organization Steering Committee
- Chair of the initial Electrical Harmonization effort
- Chair of Tools & Technology Council of the ANSI Harmonization of Product Data Standards Organization

NIST's contributions, the technical development of IGES and other foreign national standards efforts, the yet-to-be-determined success of standards harmonization efforts, and the development of STEP are inseparable from the development of the relationships among the contributors. Figure 2-5 graphically portrays the historical sequence of events of the many contributing product data standards activities.

Historical Overview

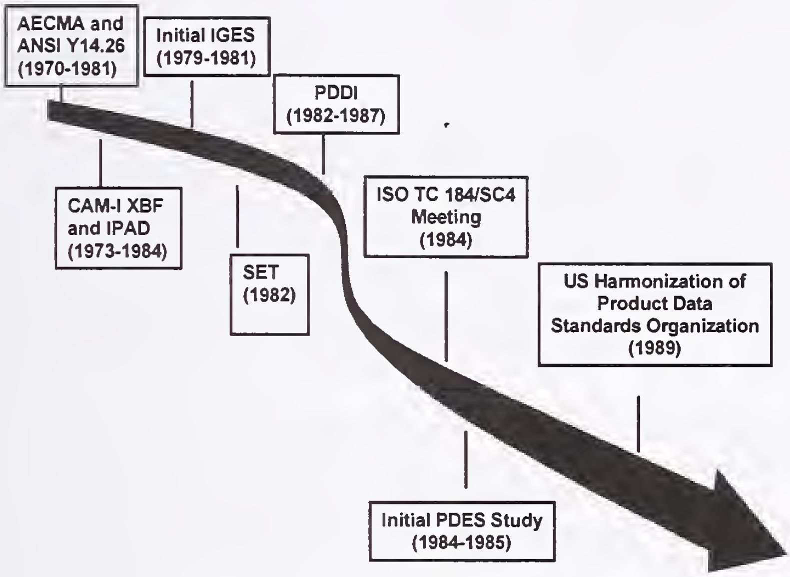


Figure 2-5: History of Product Data Standards

There was tremendous excitement about embarking on new territory; engineers were liberated by their employers to delve into research and development. Passions ran high. Vendors learned early that by opening up their systems to the public they could more readily catch a market, not lose it. Late-night conversations in smoke-filled rooms played a critical role in the birth of these early standards, as did personal trust among the participants. Once feasibility was shown through STEP predecessors, the tremendous need within industry for a formally-standardized CAD exchange capability drove the world to develop STEP. No one in 1984 could have comprehended the magnitude and longevity of the events about to unfold.

CHAPTER 3

STEP DEVELOPMENT -- CYCLES OF CONSENSUS

3.1 INTRODUCTION

ISO 10303, informally known as STEP,[8] has developed from a group of people popularizing emerging ideas through cycles of consensus, much as do schools of art, literature, and music. Such ideas eventually become the status quo followed by other groups of people with new or recycled ideas that become the status quo and so on. Chapter 2 portrayed the formal and informal dynamics of such group gatherings and their ideas. These cycles of consensus saw early domination of the U.S. with its PDES initiative. Initially the distinction was between PDES in the U.S. and STEP in the international community. Later distinctions were made exclusively within ISO TC 184/SC4 working groups. Throughout its history, SC4 has used its organizational structure as a means of shifting emphasis in developing STEP and other PDE standards.

This chapter presents a simplified view of the consensus cycles in developing STEP and establishes an historical context [20].[9] ISO 10303 is a standard whose heritage can be traced back over twenty years (see Chapter 2 for STEP's foundation). It originated with the need for geometry among simple drawing systems. As ISO 10303, STEP delivers the capabilities of all preceding related national standards, but in one cohesive package, albeit, a large package! STEP provides broad capabilities beyond our national IGES-focused capabilities. The most significant difference is that STEP not only focuses on what the data are, but what the data mean and how the data relate to each other.

3.2 THE BEGINNING OF STEP

On July 11, 1984, the first ISO TC 184/SC4 meeting was held at NIST. Participating countries included Canada, France, Germany, Switzerland, the United Kingdom, and the United States. The reason for this meeting was to create an international standard that enabled the capture of information comprising a computerized product model in a neutral form without the loss of completeness and integrity, throughout the lifecycle of a product. Several resolutions [21] described the mission, goal, and objective of this new effort:

[8] STEP describes the collection of standard documents published under ISO 10303. The acronym often appears with one of two different references. The initial referent was the "STandard for the Exchange of Product model data." The idea of product model was derived from the use of computer aided design (CAD) systems, wherein, the collection of data at any given time was the product model. The second reference was a philosophical switch to the "STandard for the Exchange of Product data." Here the emphasis was shifted from an exchange of an entire product to product data (implying that any amount of product data could be exchanged). It is the second reference for STEP that is used here, although the word "model" remains a part of the acronym. "Model" came to refer to a schema (i.e., a representation of information requirements using a formal descriptive language, rather than actual data instances as in a product model).

[9] A portion of this chapter, especially between the period 1984-1990, is quoted directly from Danner, W.F., A Proposed Integration Framework for STEP (Standard for the Exchange of Product Model Data). Natl. Inst. Stand. & Technol. (U.S.) NISTIR 90-4295; 1990 April.

RESOLUTION 1: (Gaithersburg - July 1984)
SC4 recognizes the need for a new standard for the external representation of product model data. This standard will be based upon existing data exchange initiatives including the U.S. IGES and PDDI, the French SET, the German VDA/BDMA-FS, and the UK NEDO[10].

RESOLUTION 2: (Gaithersburg - July 1984)
The SC adopts the following goal and objective:
Goal: The creation of a standard which enables the capture of information comprising a computerized product model in a neutral form without loss of completeness and integrity, throughout the lifecycle of the product.
Objective: A draft proposal for Version 1 is required for ballot by SC4 by the end of 1985.

Another resolution established NIST as a significant participant, contributor, and leader in the effort to develop STEP.

RESOLUTION 5: (Gaithersburg - July 1984)
ISO TC184/SC4 thanks Mr. Bradford Smith [of NIST] for providing his excellent work as chairman of this meeting and proposes to nominate Mr. Smith as Chairman of ISO TC184/SC4 for a three-year term.[11]

Resolution 1 identified several efforts already underway or established as national standards around the world. All of the efforts were significant, and each contributing program or country was motivated by the common good of all to meet product description requirements; however, each approach was significantly different from the others. After considerable effort at developing a consensus using existing work, the focus shifted.

The IGES Organization in the United States decided to focus on accomplishing Resolution 2. The participants at IGES Organization meetings concerned with this effort included experts from many of the countries supporting the ISO development of STEP; however, in the U. S. the new standard was identified as PDES (Product Data Exchange Specification). To emphasize this new direction, the IGES Organization became the IGES/PDES Organization (IPO). Consensus shifted toward developing PDES from scratch rather than continuing to enhance IGES. The intent of such a philosophical shift was to use state-of-the-art data modeling techniques. Eventually, PDES was proposed formally by the United States to ISO TC 184/SC4 to serve as the master draft of ISO 10303.

William Burkett recalls. . . I remember well, sitting in that small, hot hotel conference room in New Orleans, Wednesday night, May 2, 1984, about 11:00 pm or later, in a meeting of the IGES Edit Committee. In response to the challenges raised against IGES, Phil Kennicott, Chair of the Edit Committee, asked Kal Brauner (Boeing) to research and prepare a report on the "next generation of IGES." This, for me, was the "start" of STEP.

Resolution 5 established NIST as a leader in the effort as the SC4 Chair. NIST also served as SC4 Secretariat (on behalf of ANSI). NIST ended up as the Chair and Secretariat for SC4 because it suggested to ANSI, through Brad Smith, that the subcommittee be formed. The United States was named the SC4 Secretariat by the Technical Committee 184, upon NIST's suggestion, and ANSI appointed NIST to do the job. At this point in history, no one expected NIST's leadership and active participation to be necessary for more than a decade to build a successful product data standard along side industry and academia. At this point in history, no one was able to comprehend the

[10] The United Kingdom National Economic and Development Office (NEDO) ran an initiative in the early 1980s supported by Susan Bloor of Leeds University, and others to sort out data exchange in the construction industry. It was later folded into the overall national effort, but published a couple of reports and sets of guidelines which led to the formation of the CAD-CAM Data Exchange Technical Centre (CADDETC) mentioned later in this document.
[11] Mr. Smith, from NIST's Manufacturing Engineering Laboratory, served as the Chair and Secretary until October 1995.

sheer magnitude and complexity of a product data standard! Fresh, ready, and eager, SC4 began its mysterious journey to develop a comprehensive ***STEP***.

3.3 THE PDES INITIATION EFFORT

Mentioned earlier in Chapter 2, the PDES Initiation Effort [22] was a "proof of concept" project begun as an ad hoc effort within the IGES Organization to validate the methodology by which PDES would develop into a product data exchange specification. The Initiation Effort introduced the reliance on information modeling, concentrating on two aspects: formal descriptive languages and a methodology for the description of product data.

3.3.1 The Initiation Effort Architecture

> *Curt Parks recalls... I remember fondly the meeting that took place in a red brick schoolhouse on the Kansas prairie. As part of the PDES Initiation Task, several of us had been asked to assemble teams to construct domain models. When he had completed his preliminary work on the models received from the teams, John Zimmerman, as technical lead on the Initiation work, called the model's team leaders and others to meet with him to go over his "qualified and interpreted" candidate integration models. John's company had leased a vacant school in an almost rural area for activities that would not require a security clearance. Obtaining a deep understanding of electrical connections in terms of topology theory was not easy. Occasionally the meetings would drain folks to the point of dead silence. We would take a breather, gazing at our surrounds: at rolls of oilcloth maps at the front of the room and the long wood-trimmed windows which were opened with the use of a tall pole with a cast iron hook on its top. Or the glimpses from the wide hallway, past the door with the worn and waxed asphalt flooring, and lockers lining the walls. This environment provided the gentlest pull back to reality and values that only a prairie schoolhouse could evoke.*

The Final Report of the Logical Layer Initiation Task Group [23] was a major product of the PDES Initiation Effort. It contained the definition of an information model architecture with three distinct categories of models (Figure 3-1). These included discipline models, resource models, and global models. The discipline models were to capture the application-specific[12] view of the discipline experts. Resource models were to represent aspects of product description that were used commonly by multiple discipline models (e.g., geometry and topology). The resource models that formed the logical layer were to contain only generic entities and structures common to many application areas (i.e., no discipline-specific entities were to be present in the logical layer). The global models represented an informal description of the correspondence between the discipline models and the logical layer model.

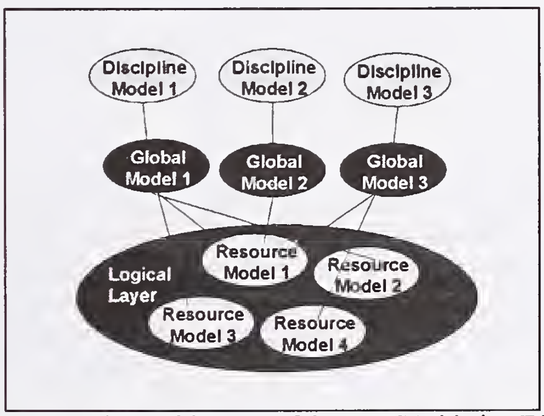


Figure 3-1: The Architecture of the PDES Initiation Effort

[12] An application in PDES and STEP refers to the use of data for a specified purpose.

The technical details concerning the development of the global models were not well understood. Therefore, in the absence of an overall plan that addressed this issue, model development proceeded independently on discipline and resource models. Discipline models were developed for applications within mechanical products; architecture, engineering, & construction (AEC) products; and electrical products. These three broad application areas were carry-over work that extended what was started as IGES initiatives. Resource models included geometry, topology, product structures, and other models that dealt with common aspects of a product's description. Most models, however, were neither clearly discipline nor resource models but rather were driven by participants who had funding to work on some particular aspect of product data representation independent of any proposed architecture. The historical approach to developing models continues to haunt the SC4 community today. It is often difficult for SC4 to establish and maintain direction and priorities. The grand experiment in building a standard side-by-side with the technology has its price.

Participation from NIST was primarily discipline-specific at this time. Participants from what is now the Building and Fire Research Laboratory contributed to work in AEC committees and led the development of the AP methodology. Similarly, participants from the Manufacturing Engineering Laboratory and Electronics and Electrical Engineering Laboratory contributed to work in mechanical and electrical product committees.

3.3.2 Modeling--What was Needed, What was Available

It is important to remember two things about models. First, a model is a re-creation or idealization of the real-world phenomenon--it is not the phenomenon. Second, a model is a simplification or abstraction of the real-world phenomenon--it does not and cannot represent all salient aspects of the phenomenon but chooses among those aspects for a given purpose.

At the outset, SC4 recognized that robust data modeling was necessary to support the complexity of STEP. Many modeling languages (e.g., ADM, ER, IDEF1X, NIAM), existed and are discussed in more detail in Chapter 5. Each was either incomplete or did not otherwise address significant areas of concern to STEP. For example, ER did not support inheritance hierarchies (although later extensions have added this concept). ASN.1[24] was the closest to an ISO official modeling standard but ASN.1 was also insufficient, lacking, for example, any means for expressing relationships. Even though insufficient, NIST recommended ASN.1 be used as a foundation to leverage developments and software tools already available from the communications standards. Why reinvent from scratch when an internationally accepted starting point already existed? NIST was a lone voice in the push to use an existing language, and lost the battle to influence SC4 in this direction. The first of perhaps many utterances from the SC4 community prevailed: "we are different, we are more complex."

The PDES Initiation Effort did not identify a single formal descriptive language. NIAM [25] [13] and IDEF1X [26] (Integration DEFinition 1X) were leading candidates for modeling languages already in use. Both of these languages used graphical syntax. A new effort also began to develop a computable language (i.e., one with a lexical rather than graphical syntax) that was to become EXPRESS [27].

One essential missing, or poorly supported, aspect of existing data modeling techniques at that time was implementability. It was desirable, after investing considerable time in creating a data model, to automate the checking and conversion of the model into software that could manipulate it intelligently. This was not generally possible. For example, consider the common problem of textual descriptions that denote constraints or algorithms that were not expressible directly in the modeling language. Most systems allowed such comments to be attached yet there was no automatic way of converting these attachments to computer software since they were intrinsically unimplementable.

---

[13] Object Role Modeling (ORM) is the current name for what was then called NIAM (Nijssen Information Analysis Methodology).

Programming languages and database languages (both for modeling and querying) were also considered insufficient. Programming languages in particular were of interest because languages such as Ada [28] and C++ [29] seemed to be approaching the power needed for modeling; however, their very nature, as implementation languages, conflicted with their use for creating abstractions that omit inessential details. On the other hand, programming languages and database languages were real languages, some already standardized or in the process of being standardized. The idea of piggybacking SC4 efforts would have had significant savings in time and labor.

Instead, a large amount of time was spent discussing the tradeoffs of both modeling languages and programming languages. Because this discussion was never recorded in sufficient detail, the same topics would be revisited periodically to no productive end. It paid off in the short term not to record technical rationale with real-world context. As is typical, investing in such documentation would have paid off over the long term but this is only painfully obvious in hindsight.

From a discipline perspective, the AEC models tended to be in NIAM; electrical and manufacturing discipline models tended to be in IDEF1X; and the geometry and topology resource models used the ever-changing lexical form of EXPRESS. National interests further exacerbated this discipline split–the United States used predominantly IDEF while the Europeans used predominantly NIAM. Within the United States NIAM users were primarily from the academic community; IDEF users were from the defense community. The scene was set for what was to become known as the holy wars of STEP over its use of data modeling languages.

NIST staff participated on two levels regarding the modeling languages. Some participants were involved as technical "language development" experts contributing to developing EXPRESS. Others were technical experts in information modeling whose focus was on ensuring the ability of a formal description language to facilitate the capture of the semantics of information requirements.

NIAM is rooted in linguistics and, therefore, was transformed easily into English statements. The NIAM graphical syntax is particularly rich in its ability to represent constraints. IDEF1X, developed with more of an orientation toward database technology, is particularly well suited for developing relational (and other) database implementations. In the area of computer programming, EXPRESS was seen by many to be developing from the roots of abstract data types. By many, EXPRESS was also considered the only lexical language that could satisfy the requirement for a computable representation of information requirements.

Each of the three leading modeling languages had its adherents. NIAM, IDEF1X, and EXPRESS were three different solutions to the problem of representing information requirements formally. Some groups tended to prefer NIAM and IDEF1X believing that these languages were designed for use at the conceptual level to capture semantics free of implementation considerations. They saw EXPRESS as a syntactic approach, appropriate to a logical level description that included implementation considerations. This was due largely to its explicit declaration of data types. EXPRESS was expected to require usage conventions to be more precise about the semantics. The fact that the PDES Initiation Effort had used the term Logical Layer only added to the confusion, since what they described was seen essentially as conceptual in nature. Had both conceptual and logical models been developed in ISO 10303, many difficulties with implementation might have been less severe.

EXPRESS also included a procedural constraint specification capability that was less desirable in the eyes of many information modelers than the declarative constraint approaches in NIAM and IDEF1X. Others, however, preferred the syntactic precision of the data type representation and procedural constraints in EXPRESS. Therefore, models developed for STEP reflected diverse information requirements at many different levels of abstraction, determined by those who had funding to work on the developing standard, and also reflected representation preferences with respect to these three formal descriptive languages. In retrospect, might another data modeling language have been a better choice? Perhaps a better choice would have been a more traditional programming language such as C++ (although C++ was not stable at this time either).

At the time the STEP activity began, participants had an unrealistic expectation -- that it would be possible to create a satisfactory modeling language in only a few years, including standardization and acceptance. One might ask, how could anyone be that naive? Perhaps it was not naivete so much as an earnest drive to build and complete a master product data exchange standard in a timely fashion.

Some of the contemporary modeling languages also improved over the years and it is possible that STEP developers could have focussed on one of these existing languages that was "close enough" such as IDEF1X. EXPRESS ultimately evolved from McDonnell Aircraft's PDDI program (a PASCAL derivative known first as RASCAL and then DSL (Data Specification Language or, as some humorously observed, the Doug Schenck[14] Language). In retrospect, PDDI was not the right place to start. The current work bears little resemblance to the PDDI contributions.

Another problem was the background of the participants. At NIST, one was not a modeling expert; one was primarily an engineer. Much of our time was spent in self-education. We had very little expertise in creating robust modeling languages. For those who could model, there was not enough modeling expertise to go around. Since NIST led the technical effort via the Chair and Secretariat positions, it had a certain amount of ability and responsibility to affect the direction of STEP's modeling standardization. The skills, understanding, and background of NIST's participants were also representative of the majority of other participants at this time. Thus, a little of the "engineers leading the engineers" occurred. It created a resource management problem that plagued the activity for years.

3.4 THE INTEGRATED PRODUCT INFORMATION MODEL (IPIM)

By October 1988, the information models developed within the PDES and STEP projects had been assembled into a single model, the Integrated Product Information Model (IPIM). This represented a shift away from the IGES Organization's suggestion in its PDES Initiation Effort Final Report. The shift now viewed models of whatever type to be translated into EXPRESS and then combined into a single entity pool from which implementors could draw for effective data exchange. The IPIM was developed within ISO TC 184/SC4/WG1 Subgroup (SG) 6 on Integration; however, SC4 consensus seemed to be shifting to the ideas presented by another SC4/WG1 subgroup that would result in quite a different architecture.

3.4.1 The IPIM Architecture

The IPIM was the grand "Big Daddy" -- the summation of all models (represented in EXPRESS) regardless of their level of abstraction (Figure 3-2). Due primarily to the entity pool concept adopted for the IPIM's specification, every model or entity could serve potentially as a resource to any other model. The entities could be drawn upon on an ad hoc basis to create new models. Models effectively established partitions within the IPIM between what was considered relevant to a given objective, and what was not. This entity pool approach provided considerable flexibility for model developers; however, it required careful attention to issues of integration, and the creation and contents of the partitions. Models developed in relative isolation of one another could create multiple, and potentially conflicting, ways of accomplishing the same objective. What was flexibility for the modelers could easily become ambiguity and redundancy for the implementors and users!

[14] Doug Schenck was Chair of the SC4 working group that initiated development of EXPRESS.

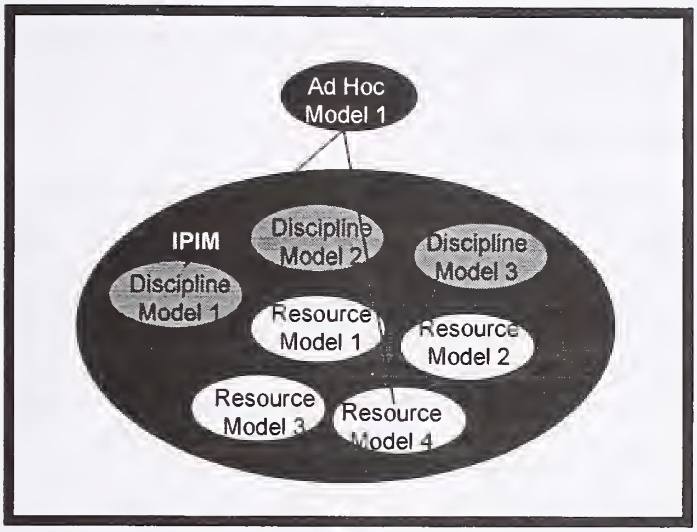


Figure 3-2: The IPIM Architecture of ISO TC184/SC4/WG1

Integration in the IPIM was limited to combining the "surface structure" of the models. That is, the meaning of the entities was reviewed in a literal sense defined by the modelers. If two models used different names for the same object, or used the same name for different objects, a naming conflict existed that required resolution. Although it became evident that an analysis of the underlying meaning of concepts was needed, it was not undertaken. Therefore, conflict resolution was not required to resolve differences, for example, between the AEC and electrical disciplines' modeling of connectivity because generic entities would have been defined that were common to both disciplines.

Jim Nell recalls... In the middle of the integrated-resource wars we were continually trying to fix on consistent meanings for entities. The UK team of Howard Mason and Nigel Shaw distributed the following cricket rules at the meeting prior to Heathrow to highlight the integration problem. (I wonder if a machine could parse a product file such as this.)

<u>Ins and Outs of Cricket</u>

Two teams of 11 players toss a coin to see who goes in first. Team that is not in goes out.

First two players of the team that is in, go out. These two stay in until one of them is out then he comes in and another player that is on the team that is in goes out.

This goes on until only one man is left in, then they are all out, apart from the man who is not out and both teams come in.

Then the team that was in goes out, and the first two players of the team that are now in go out.

Scoring appears to be unimportant, but the team who runs the most wins.

Often it rains. Then both teams come in and depart to the inn.

(Who said integration was complicated?)

Two principal criteria were applied in the entity level review of the IPIM. The first was schema independence of entities: each entity name was unique throughout the IPIM. The second was context-independence of entity constraints: each entity included only constraints that were independent of context. Both of these criteria maintained the ability to use any combination of entities in developing discipline-specific models (i.e., modularity was the governing consideration). Other criteria included minimal redundancy, but time and resource limitations prohibited their thorough use.

Information models that were being developed from the viewpoint of a particular discipline were referred to as application models. Resource models were those with capabilities used across application models; however, this distinction could only be made in a relative sense because few application models used other resource models. The distinction did not seem to be particularly important to the IPIM approach. The formal distinction between application models and a logical layer model with correspondences established between them had apparently been abandoned.

3.4.2 U.S. Activities at This Time

During the mid-1980s and the PDES Initiation Project, Kal Brauner (Boeing) was the PDES Project Manager in the IPO. When he resigned in the mid 1980s, Thurber Moffet (Northrop) took over. When he assumed the PDES Project Leadership, he actively pursued the creation of an independent U.S. organization to focus solely on accelerating the development of PDES. The fruit of his labor was PDES, Inc. PDES, Inc. undoubtedly has had a very marked impact on STEP during the formative years of the standard (1988-1992) and continues to have an impact today. A description of PDES, Inc. can be found later in this chapter. It was also during this time that the teaming effort of Phil Kennicott and Peter Wilson began developing the concept and content of the IPIM.

3.4.3 The Formal Description Language of the IPIM

The IPIM used the emerging EXPRESS data specification language, which meant that the models were being modified continually as EXPRESS was also being developed. The IPIM finally used what was called the frozen version of EXPRESS [30] for the explicit purpose of IPIM documentation and the first SC4 ballot that became known as the Tokyo Draft [31]. The Tokyo Draft was approved, by SC4 Resolution 29, for registration as an ISO Draft Proposal (DP). At this time, the following parts were recognized as part of the initial draft:

- ISO TC 184/SC4/WG1 N279 (Physical File Structure)
- ISO TC 184/SC4/WG1 N280 (Mapping from EXPRESS to Physical File Structure)
- ISO TC 184/SC4/WG1 N283 (Introduction, Scope and Definitions)
- ISO TC 184/SC4/WG1 N285 (Test Methods)
- ISO TC 184/SC4/WG1 N287 (EXPRESS)
- ISO TC 184/SC4/WG1 N284 (IPIM) with the exception of the user-defined-entity

SC4 directed NIST, as its Secretariat, to circulate the DP for letter ballot according to ISO Directives. The ballot was to include voting on each clause and each section of Clause 4 of the DP. The initial draft ballot was open for three months. This ballot started the ISO "clock" to produce an international standard in seven years. Ironically, with such a landmark occurrence in the SC4 community, the United States was torn in support of the then current version of PDES going forward for SC4 ballot. Many in the United States' Technical Advisory Group (TAG), particularly the NIST TAG representatives, felt PDES still needed to be more complete technically before it merited international scrutiny. On the other hand, many felt the need to get PDES/IPIM into the SC4 community before another entirely different approach was presented by another country. After a grueling five hour TAG meeting, the guidance given to the

Sharon Kemmerer recalls... The Tokyo meeting has the most historical significance because of this landmark resolution. It also had a humorous (to some!) start. I arrived in the hotel lobby after almost 24 hours spent in airports and flight. The hotel clerk, noticing my American accent, pleasantly greeted me with a message: Jerry Weiss, the WG1 Convener, was not able to leave the United States and someone else would have to run the week-long meetings. It seems he got to New York from his origin in Texas, only to learn that he needed, and did not have, a Japanese visa! So, upon depositing my bag in my room, I ran back downstairs to gather any leadership-type bodies that may be hanging around in the lobby. My search yielded people like Nigel Shaw and Howard Mason from the United Kingdom. We met there in the lobby, wrapping up close to midnight with an agenda in the hands of our newly appointed ad hoc leader: Howard Mason. It proved to be an interesting start to a very interesting meeting.

United States delegation heading for Tokyo was to vote "no" on the above resolution. If one would check the history books for Resolution 29's country vote, however, one would note the United States voted "yes." Although this change in the prescribed vote was not received popularly back home at the time, we now have the gift of hindsight. The United States vote of "yes" to support "letting loose" the PDES draft was technically and politically crucial and appropriate at this point in time. It showed U.S. solidarity with the rest of the world to create one international product data standard.

3.5 CONTEXT DRIVEN INTEGRATED MODELS (CDIMS)

The National PDES Testbed at NIST and PDES, Inc. (both described later in this chapter) set out to test the validity of the IPIM through the development of prototype implementations. Both organizations soon concluded that implementation of the entire IPIM was not only difficult, but provided multiple contradictory approaches to representing the same application requirements. PDES, Inc. led the way to develop the idea of Context Driven Integrated Models (CDIMs) in 1989 to pursue the further development, implementation, evaluation, and validation of PDES from an application context view. CDIMs were developed to identify useful portions of the IPIM for explicit purposes. They were the pre-cursor to APs, and the methodology for developing CDIMs led to a better understanding of APs. A CDIM defined an explicit application context and the subset of the IPIM that would satisfy the information requirements of that context. The introduction of the CDIM concept began a movement away from the flat single entity pool architecture back to something more similar to the PDES Initiation Effort architecture (Figure 3-3). The first CDIM developed had as its scope, the exchange of 3-D product design data in a configuration controlled environment. This CDIM evolved into one of the first two ISO 10303 APs: 10303-203.

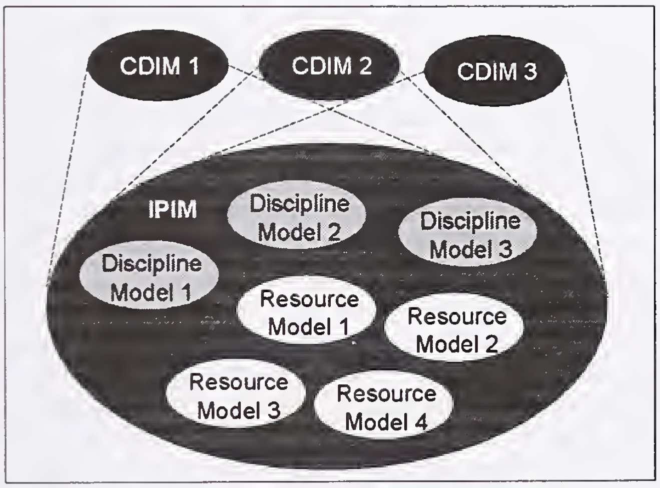


Figure 3-3: The Architecture of U.S. Implementation Testbeds

3.6 THE INTEGRATION MODELS OF THE PDES INTEGRATION TASK GROUP

Under the auspices of SC4/WG1/SG6 the IPIM was being developed and ongoing testing efforts were underway to develop application context subsets to implement the IPIM. At the same time the IPO formed an Integration Task Group (ITG). This Task Group dealt with the semantic integration of selected models that were considered true "resource models." Although the United States' "PDES" was now under the international umbrella for development, the U.S. was somewhat politically hesitant to drop all its PDES development work. The U.S. was still in its infancy of understanding product data representation – like a child thrilled with each discovery of fingers and toes. Besides, millions of industrial and government dollars were already invested in PDES; marketed solutions were already touting PDES; and very visible entities with PDES in their name (e.g., PDES, Inc., IPO, National PDES Testbed) currently existed. When so much is built around an acronym everyone knows what has to be done:

keep the acronym but change its meaning. More than a year after PDES was embraced by SC4 resolution 29, "PDES" became Product Data Exchange using STEP. Now politically correct, the U.S. technical commitment to a single international standard was re-affirmed.

The ITG began the task of integrating several of the discipline models at the model development level. They chose models that were at a level of abstraction that seemed appropriate to resource models that could be used in many application contexts. Although the United States took the leadership in this activity, the results were being fed back continuously into the SC4 community.

MIKE PRATT RECALLS... in 1984, when Brad Smith gave a talk in Europe about some IGES data transfer experiments. The final stage of the experiment was the manufacture (by machining) of the modeled part. Brad commented on the size of the part, and said that it was actually machined half-size - even so it was some 18 inches long. The audience laughed heartily at this, and Brad was totally taken aback. But they mostly knew that the part concerned belonged to an aircraft door locking mechanism. It was dimensioned in millimeters, not inches, and the real part was only about 1.5 inches long! I believe that STEP avoids this problem by demanding that units be specified.

Application and resource models were candidates for integration. Models to be integrated were chosen based on both model development status and stability. Six models were chosen for consideration. They included Product Structure and Configuration Management (PSCM), Finite Element, Materials,[15] Tolerances, Form Features, Geometry, and Topology. The Shape Representation Interface Model had been completed by January 1988 and was therefore part of the IPIM. The distinction between application and resource models made by the PDES Initiation Effort had been reaffirmed by the PDES ITG work efforts. The resource models provided the required functionality for defining the shape of a product in terms of its representation. An application model was concerned with the definition of a product.

By July 1988, the PDES ITG had become the Integration Committee of the IPO. In January 1989, it formed two subcommittees, Integration Resource and the Integration Practice. Integration Resource served as a forum for the PDES project to discuss technical issues regarding the integration of generic resource models. It was responsible for developing a strategy for integration. Integration Practice executed the strategy developed by the Integration Resource. Integration Practice included modeling experts and members of the model development committees who were charged with the responsibility of acting as experts on particular models during the integration process. Integration was to take place in small working task groups. These IPO subcommittees were functionally analogous to SC4/WG4. The United States provided continuity across IPO and SC4 work by convening WG4.

Early in the discussions held by Integration Resource, it became evident that models in the IPIM varied along a continuum of generalization (i.e., the degree to which they included generic concepts rather than concepts specific to any given application). Some models were very specific, such as the Ships Structural Systems Model. Others were more generic, such as the General AEC Reference Model (GARM), the Electrical Functional Model (EFM), and the PSCM Model developed by experts from the AEC, Electrical, and Mechanical products disciplines respectively. Multiple approaches to specifying generic aspects of a product were also developing. The PSCM was generic in nature, focusing on a clear specification of product structure and its configuration management ramifications. The GARM was generic in nature but had a different approach to modeling product structure. In addition, the GARM included general product characterization and many entities that resulted from its consideration of product lifecycle.

The integration of the very specific application models with the more generic models was unclear and inconsistent. Within the AEC Committee, for example, the integration of a specific model like the Ship Structural Systems Model with the GARM was proceeding slowly or not at all. A methodology for integrating models at different levels of generalization was absent. This suggested that the function of establishing correspondences between specific and generic representations of a product by something like the global models of the PDES Initiation Effort was still relevant.

[15] The original Finite Element Model (FEM) dealt with materials. The Integration Task Group suggested a division into a Materials model and an FEM.

3.7 ISO RECOGNIZES THE CONCEPT OF APPLICATION PROTOCOLS (APS)

"A rose, by any other name will still smell as sweet..." [William Shakespeare]

When the STEP project was initiated in 1984, only a general description of its intended scope existed. That scope included the representation and exchange of all product data necessary to define completely any product for all applications over the product's entire lifecycle.

By the Tokyo meeting in December 1988, SC4 recognized the need for application *profiles* and passed Resolution 34. SC4 resolved that application profiles of STEP shall form separate parts of the STEP standard, in order to provide a basis for conformance testing. The use of the phrase "application profile" was a short-lived attempt to parallel ISO/IEC JTC1's[16] use of the phrase application profile. The exact phrase to use was actually a point of technical debate within NIST; however, consensus ultimately was achieved within NIST, and NIST carried the "banner to correct" to the SC4 Paris meeting in January 1990. Resolution 54 reflects the cementing of the phrase application protocol: "The term 'Application Profile' identified in SC4 Resolution 34 is to be replaced by the term 'Application Protocol'."[17]

At the June 1989 ISO TC 184/SC4/WG1 meeting in Frankfurt Germany, application protocols (APs) [32,33] were acknowledged to serve an important role in determining how STEP would proceed. The concept of an AP, introduced in Chapter 2, had been developed within the IPO Application Validation Methodology Committee. Its purposes were to 1) state explicitly the information needs of a particular application, 2) specify an unambiguous means by which information is to be exchanged for that application, and 3) provide a basis for standardized conformance verification.

Where application protocols fit in the ISO 10303 architecture is presented in Chapter 4. Today's elements of an application protocol are detailed in Chapter 7.

Adopting the AP methodology essentially reestablished the basic architecture of the PDES Initiation Effort. One would note this was roughly two years after the PDES Initiation Effort basic architecture was developed, and the STEP developers had accomplished a 360-degree change! This is one of the many lessons learned in the breaking of new ground in standardization practices. ISO TC 184/SC4 was attempting to standardize the state-of-the-art, the unknown, and the unstable. The negative outcome, of course, is inefficiency, repetition, and time-delay. The positive outcome of such ambition is the potential for a more robust standard, existing implementations when the standard is released, and a better technical understanding of the elements comprising the standard.

An additional benefit suggested from the experience is, "he who gets the first working draft, benefits most." On more than one occasion the technical directions recommended in the PDES Initiation documentation contributed heavily to the ultimate direction agreed upon by SC4.

Application Reference Models (ARMs) were application-specific models with clearly defined scopes and functional requirements. This constituted a refinement of the original discipline model concept: resources were to be used in an indefinite number of application contexts. They were generic models used to provide the required functionality of APs. The mapping tables (MTs) and Application Interpreted Models (AIMs) were intermediate representations (refining the concept of global models) that provided a formal description of the mapping between the ARMs and the resource models. They specified how the resource models were to be used for particular applications in terms of constraints for populating implemented resource models. The informal description of correspondence identified by the Initiation Effort had been refined by the AP methodology (Figure 3-4).

[16] ISO/IEC Joint Technical Committee 1 is responsible for standardization in the field of information technology.
[17] The naming convention for "application protocol" is based on the definition of application protocols in ISO 9646, Open Systems Interconnection suite of standards.

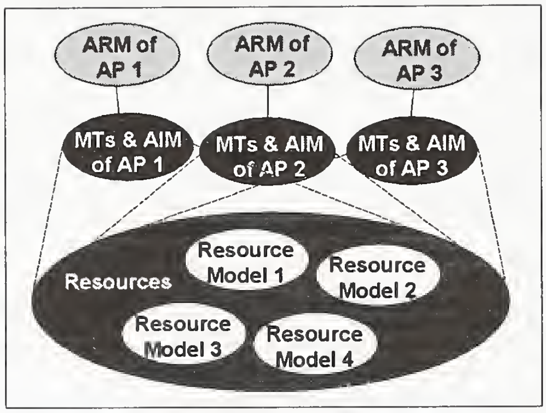


Figure 3-4: AP methodology refinement of the PDES Initiation Effort Architecture

The AP methodology emphasized explicit and well-documented elements for an application protocol. However, two significant outstanding issues remained. The first arose from the incomplete understanding of the technical details of the process by which equivalence was to be maintained in APs (i.e., correct interpretation in developing MTs and AIMs). The second was that the resource models of the IPIM had alternative ways of representing the same information into a single entity pool that resulted from the uncoordinated development approach and a lack of semantic integration during the IPIM assembly.

The solution: each application protocol could develop a unique way of achieving its ARM. Although the focus was still to maintain consistent AIM structures, the potential for separate application protocols that were fundamentally incompatible highlighted the consequence associated with ad hoc development. The development of compatible rather than "peacefully coexisting" (i.e., incompatible) application protocols was desirable. Compatible APs could be merged and altered as information requirements changed. Thus, a single coherent representation of common product data was required, and the AP methodology seemed to suggest the possibility of a sensible resolution to disputes over modeling languages as well.

3.8 SO CAN WE BUILD A STEP PLANNING MODEL?

The PDES and STEP projects made numerous attempts to develop a planning model that would represent coherently product data common to multiple applications. By early 1989, considerable progress had been made by ISO TC184/SC4/WG1 SG5, the Data Architecture Subgroup. Criteria had been established for a planning model, but there were nearly as many planning models as participants in the Data Architecture Subgroup! Each of these planning models had been developed as a top-down approach to STEP development (as contrasted with the collection of models contained within the IPIM that had, for the most part, been a bottom-up approach). The General AEC Reference Model was renamed the General Application Reference Model (in the spirit of keeping the acronym, just changing its meaning) [34]. It appeared to meet the minimum criteria of SG5.

The planning model was presented as a first step toward explicitly stating the scope and nature of the generic information requirements of the PDES and STEP projects. As such, it could be used to analyze potential resource models, to identify areas of strength and weakness, and to plan a strategy for future development.

The same fundamental structure was developed independently by two groups, one under the auspices of ISO TC 184/SC4 and one under the auspices of the Integration Committee of the IPO. These two groups presented a model classification and planning group (working top-down) and a model integration group (working bottom-up). The merger of these two groups formed the basis for building consensus that technically differed significantly from the IPIM.

3.9 THE GENERIC PRODUCT DATA MODEL (GPDM) OF THE PDES INTEGRATION RESOURCE SUBCOMMITTEE

By October 1989, the IPO PDES Integration Resource Subcommittee used deep-structure integration as a means of uncovering fundamental concepts within product data models. The term deep structure integration draws by analogy from a distinction made in linguistics between surface- structure and deep-structure representations of meaning [35]. The potential resource needed to be examined both in terms of the surface representation of the particular discipline for which they had been developed, and in terms of more fundamental underlying concepts applicable to products in general. Deep-structure integration was the means by which fundamental concepts could be identified.

Curt Parks recalls... I remember the informal event to win the computing hardware goodies contest in 1989. Many of us were working on STEP models, tools, and documentation. Most tasks required better computers than we had been routinely using. (Those old VAX 1170 terminals presented a problem when it came to EXPRESS-G models.) Some of us put in for new computers, which often came with "nice" extras. In those days, "nice" meant color monitors, stereo sound, video, and CD-ROM players! With a choice acquisition often came comments of "goodies-gathering." No one, however, ever topped Cita Furlani's installation of a donated mainframe Tandem Computer system. This large-size non-stop data cruncher did not do mundane things like serving mail; it was dedicated to handling BIG loads of STEP data. It could eat our PCs for breakfast! No one else got hardware that came remotely close.

3.9.1 The GPDM Integration Architecture

The results of the deep structure approach to integration were consistent with previous work that had reaffirmed and refined the architecture of the Initiation Effort. The integration framework that emerged had the GPDM as its central feature. The GPDM captured common elements of product data in a single coherent representation. It provided an application context-independent description of a product in terms of generic product description facts (i.e., facts that apply to any product). It served as the missing piece to using the AP methodology in the STEP architecture! The GPDM product description facts also served as the interpretable resource elements for mapping tables and AIMs.

The GPDM provided a structure for the models that served as resources for application protocols. These models were integrated into what became known as the Integrated Resources (IRs). (Figure 3-5).

As components of an AP, application interpreted models (AIMs) formally described the interpretation of generic facts about a product in a specific application context. They made use of lower level of resource models through the GPDM. Application reference models (ARMs) had access to the GPDM by way of mapping tables and AIMs.

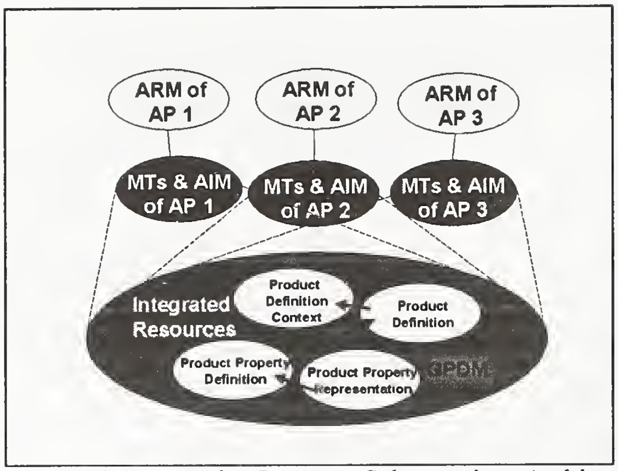


Figure 3-5: The Integration Resource Subcommittee Architecture

The product-description-facts concept of the GPDM (which were necessary elements of the IRs) developed in the IPO Integration Resource Subcommittee, corresponded to the types of product description data identified in the Data Architecture Subgroup of ISO TC 184/SC4. A merging of the minds continued to develop…

3.9.2 Consensus on Formal Descriptive Languages

The collective use of formal descriptive languages across the developing STEP architecture varied. Table 3-1 shows language use at this time in history.

STEP Architecture Component	Role of Component	Descriptive Language Used
AAM	to capture activities to be supported by the AP	IDEF0
application objects and assertions	to specify information requirements of the AP	English
ARM	to facilitate understanding of the AP information requirements	IDEF1X, NIAM, EXPRESS-G
AIM	to capture the activities and information flows to describe the interpretation of generic facts about a product in a specific application context	EXPRESS, EXPRESS-G
Mapping Tables	to trace the application information requirements	Mapping Table syntax (derived from EXPRESS)
GDPM	to capture in a single coherent representation, common elements of product data	EXPRESS, EXPRESS-G
Integrated Resources	to define the resource constructs for a particular AP	EXPRESS, EXPRESS-G

Table 3-1: Modeling Language Application

The ISO TC 184/SC4 effort started a single working group (WG1). Under WG1, it grew subgroups as the need arose. This structure included SG5 (Data Architecture) and SG 6 (Integration). SG5 was taking a top-down look at the STEP architecture using a planning model and model classification scheme. Entirely independent work was done in SG6 that produced the IPIM (i.e., Tokyo Draft). The results of the ISO ballot on the Tokyo draft showed little support for the IPIM; however, a consensus on the direction of STEP had been recognized. WG1 acknowledged the importance of APs. The U.S. thoughts merged with those of SG5 regarding the types of product description data that should be both covered and integrated. EXPRESS became accepted as the language to be used for the IRs and for the AIMs. Only two new EXPRESS constructs were needed to make the methods for integration and interpretation possible. With these new constructs, a consensus was possible for the Initial Release of STEP. To affirm this new consensus, it was at this point in history that the IPO redefined PDES to mean Product Data Exchange using STEP.

William Burkett recalls... I remember the exact moment when PDES changed from PDE- Specification to PDE - using STEP. It was the IPO Steering Committee at the Anaheim Convention center in 1990. In discussing the relationship between PDES and STEP, Spencer DePauw (from Caterpillar) leaned over to me and said jokingly "maybe we should call it Product Data Exchange using STEP!" The Steering Committee thought it was funny - and a good idea, too! They immediately voted to adopt the new meaning for the acronym PDES.

3.10 THE COMICAL, THE TECHNICAL, AND THE INSPIRED

Concurrent with the evolving understanding and adoption of modeling languages, modeling methods, and supporting architectures, were many other initiatives that contributed to STEP as we know it today. The following activities are offered to show the zeal of the participants and the magnitude of investment in this product data standardization initiative. As you will note with the Ad Hoc Complainers and Gripers Committee, it is important that one maintains a humorous outlook when developing standards. Nevertheless, even with humor, technical issues were identified and attacked with vigor. The many user communities were inspired by the scope of STEP conceptually, and were willing to put their money where their requirements were to make STEP a reality.

3.10.1 A User's Leveraged Influence--U.S. Department of Defense

User-driven requirements have proven to be a powerful motivator for developing ISO 10303. As mentioned earlier, this approach sometimes had its price because resources were driven to support particular domains of need, independent of whether or not an integrated planning model existed (see next section). On the other hand, without user requirements we would have no need for a standard. The U.S. Department of Defense was an early player and a strong player. It recognized its requirements and was not afraid to sponsor solutions to meet those requirements. Within the product data standardization community two particularly visible defense "players" were NIDDESC and CALS.

3.10.1.1 NIDDESC

The Navy/Industry Digital Data Exchange Standards Committee (NIDDESC) was a cost-sharing venture between private firms and government organizations. The basic objectives were to develop an industry-wide consensus on product data models for ship structure and distribution systems, and on the digital exchange of product model data [36]. NIDDESC's early interests and sponsorship in the area of IGES are noted in Chapter 2. NIDDESC is also noted in Chapter 7 for its contribution and work on STEP application protocols. Many of the companies committed to the early efforts of NIDDESC still play a very active role in developing a suite of ISO 10303 application protocols for the shipbuilding domain, totally integrated with other international needs as well.

3.10.1.2 CALS

Perhaps Shakespeare's quote of "...a rose, by any other name..." is applied more aptly to the meaning and acronym of "CALS." The acronym changed 2-3 times[18] from its original meaning in 1985. The change in meaning was done purposely to capture their newfound insight of their scope of requirements for fully integrated, information-managed defense manufacturing support. The reader will see references throughout this document to CALS sponsorship, CALS participation, or CALS requirements. During the mid 1980s into the early 1990s, the Defense CALS initiative provided critical funding, resources, and prioritized requirements to drive product data exchange

[18] CALS: Computer-Aided Lifecycle Support, Computer-aided Acquisition and Logistic Support, Continuous Acquisition and Lifecycle Support, and Commerce at Light Speed (industry use). Defense issued a statement (Office of the Secretary of Defense, "CALS Definition and Vision Statement," CALS Journal, Spring 1994) to explain the transition from computer-aided to "continuous." The intent was to recognize the critical importance of information-managed manufacturing, while recognizing that all operations will continue to be "computer-aided."

standardization. Today, the defense CALS requirements are being identified and prioritized internationally at the North Atlantic Treaty Organization (NATO). CALS representatives from several NATO nations now bring their requirements to the ISO TC 184/SC4 table for development and adoption under ISO 10303.

3.10.2 Ad Hoc Complainers and Gripers Committee

At the July 1987 IPO meeting, representatives from General Electric (Peter Wilson); Boeing (David Briggs); University of Leeds, UK (Nigel Shaw); IBM (Ed Clapp), and McDonnell Aircraft (Bill Burkett) were asked to form an ad hoc committee to document PDES issues. They dubbed themselves the Complainers and Gripers Committee, and prepared a paper [37] which documented more than a dozen issues and the significant events affecting the issues. A highlight of some of these issues:

- **The continued use of PDES and STEP**. Although one will notice well after the Tokyo draft adoption in 1988, the use of the acronym "PDES" lingered on. The C&G Committee clarified for all that from a technical viewpoint "PDES" and "STEP" were no different.

- **Parsing the PDES scope**. The C&G Committee viewed the PDES scope on multiple levels. At the broadest level, the scope defined a mammoth undertaking, which would take several years to complete. At the narrower levels, the scopes offered both what are available now (the implementations scope) and what will be available in the future as well. It was important for these levels of scope to be made public so that expectations for the standard could be more realistic. Understanding these different scopes would better prepare the users for a continuing set of standard releases, driven by enlarging the target customer base and implementation technologies, and by advancing the understanding of information needs for manufacturing.

- **Having a planning model**. Although some progress had been made by the time of this report, this effort to standardize product data exchange still did not have an integrated planning model. Such a model should provide the overall standard's scope, a breakdown of the work into reasonably self-contained areas, and a high level view of the interrelationships and interfaces between the work areas.

- **Integration**. Like this document's example for preparing this book offered in Chapter 1, the C&G Committee recommended PDES be developed using scenario 4. The standard should not be written as separate pieces of work, but be integrated across its pieces to remove all inconsistencies and to fill in missing ideas that may not have been captured elsewhere. In PDES terms, this was the Integrated Planning Model (which was missing, as noted in the previous bullet) and the Integrated Product Information Model (IPIM).

- **Testing**. Each portion of PDES should be tested, or validated, before approval is given to progress to the next stage in the development process. The C&G Committee recommendations included tasks to existing committees to define what this testing means and to develop test plans and procedures; and a plea to accelerate the endeavor by developing and supplying software testing tools. You will note below and in Chapter 8 that NIST took these recommendations seriously in 1987 and still does today.

- **Exchange technologies**. The exchange technologies that could be implemented fall into two major types: file exchange and dynamic exchange. File exchange includes static file exchange as with IGES and active file exchange as with the PDDI working form. Dynamic exchange also has two types: direct database-to-database transfer and intelligent knowledgebase. The introduction of this issue led to another ad hoc initiative to discover the possibilities of implementation levels.

3.10.3 Special Group on PDES Levels

Although no official definitions or standards for the four implementation levels ever emerged, the convention of categorizing STEP systems in this manner is widely used and informally accepted today. Historically, there was much effort to further understanding of exchange technologies. In the spring of 1988, the IPO PDES Steering

Committee Special Group on PDES Levels made a call for papers to discuss the idea of "implementation levels." Several contributions were received, including those from NIST, Westinghouse, McDonnell Aircraft Company, Ontologic, SDRC, Leeds University, Computervision, D. Appleton Company, Inc., and the IPO PDES Physical File Structure/Implementation Subcommittee. NIST sponsored a workshop in May, 1988 to review all the contributions and to discuss the definition of levels, as well as to raise relevant issues.

The Special Group's concluding paper [38] provided the following working definitions for each level:

- **Level 1: Passive File Exchange**: An ASCII exchange file is mapped into or out of a native CAD database format using translators.

- **Level 2: Active File Exchange**: An ASCII exchange file is loaded (moved or converted) into or unloaded out of a local working form. This working form is manipulated and retrieved by translators or applications using access software function calls.

- **Level 3: Shared Database**: Product data will be stored in a state-of-the-art, multi-user database management system (DBMS). Applications and translators can directly access and share the data through multiple external views using the DBMS data manipulation language. It is possible to use a Level 3 implementation as the master database for a product.

- **Level 4: Shared Knowledgebase**: Product data will be stored in a state-of-the-art multi-user Knowledgebase Management System (KBMS). Applications and translators can directly access and share the product data through multiple views using the KBMS data manipulation language. It is possible to use a Level 4 implementation as the master database for a product.

Chapter 6 puts the discussion of implementation levels in context when discussing exchange versus sharing.

3.10.4 A National PDES Testbed

The National PDES Testbed hosted by NIST was initiated and sponsored by the U.S. CALS program. It was established in 1988 to support STEP development activities. The Testbed assumed a critical role in developing STEP by providing a testing-based foundation for STEP's rapid completion. Brought to closure as an initiative in 1996, the National PDES Testbed has assisted in advancing STEP development through draft specification validation, tool development–for facilitating standard's development and for testing implementations of the standard, and prototyping. NIST worked closely with the U.S. Department of Defense to identify and prioritize requirements, and worked closely with industry to build a standard to meet those requirements. Although the National PDES Testbed is no longer operating as an entity, many of the tools developed over the last decade are still in use by industry today. Chapter 9 discusses many of these tools in more detail [39].

3.10.5 NIST-Wide Product Data Exchange Task Group (PDETG)

NIST founded in the late 1980s the Product Data Exchange Task Group (PDETG) to facilitate its work in providing integrated and effective service to U.S. technical and industrial communities. The membership of the Task Group consists of technical-professionals actively engaged in product data exchange research, technical development, standards adoption, validation, and conformance testing. Members represent the spectrum of domain interests. Originally the PDETG met weekly to share information, coordinate national and international ballots, and discuss technical issues. Today, the PDETG meets monthly, has its own internal web site, and does a large portion of its information sharing via e-mail.

3.10.6 An Accelerating Initiative--PDES, Inc.

In April 1988, several major U.S. companies incorporated as PDES, Inc., with the specific goal to accelerate developing and implementing ISO 10303. Hosted at the site of the South Carolina Research Authority (SCRA), PDES, Inc. is divided into two primary groups: development and deployment. The PDES, Inc. Development Group participates in:

- Developing a STEP framework/architecture that supports application protocol interoperability and upward compatibility.
- Providing technical tools and solutions for STEP implementations (e.g., geometric accuracy).
- Extending the STEP standard to include member company-defined requirements.
- Helping maintain STEP Parts critical to PDES, Inc. member companies.

The PDES, Inc. Deployment Group's primary focus is to effect the implementation of ISO 10303 in member companies. This group currently conducts and supports several STEP pilot projects in conjunction with the member companies. Most of PDES, Inc.'s resources are focussed in this group [40].

Today, several companies (both national and international) and government agencies (including NIST) are active participants in PDES, Inc.

3.10.7 PlantSTEP, Inc.

In late 1994, U.S. industry leaders of the process, power, engineering, and construction industries worked with NIST and the Construction Industry Institute to establish PlantSTEP, Inc.. PlantSTEP is an industrial consortium dedicated to accelerating the delivery of needed international standards for exchanging and sharing information about process plants. The initial project of PlantSTEP was to develop the ISO 10303-227 for exchanging plant spatial information. With the successful completion of 10303-227 as a Draft International Standard, PlantSTEP expanded its program of work to include pilot industry projects and investigations of standards used to share engineering information over the lifecycle of process plants.

3.10.8 A National Focus for PDE Information and Activities

There were several initiatives within the United States and NIST to promote digital product data exchange (PDE) as the solution for many manufacturing processes, and ISO 10303 as the means to that end. The following highlight a few of these initiatives.

3.10.8.1 The National Initiative for Product Data Exchange (NIPDE)

NIPDE was an effort initiated in 1991, with a purposeful sunset after three years. Hosted at and administered by NIST, it served as a central focal point for information regarding the large number of organizations and programs involved in developing, testing, and implementing product data exchange standards --- both nationally and internationally. Two of NIPDE's accomplishments are:

- Developed a product data exchange library on the world wide web.
- Produced a STEP video to raise the awareness of the industrial community to the significance of STEP [41].

NIPDE provided a successful demonstration of industry-government collaboration. During its three years of existence, it helped put STEP and its purpose on the map for U.S. industry. STEP took on a much higher national presence assisted by the high level of government and key industry executives serving on the NIPDE Board. The Department of Commerce Undersecretary for Technology Administration chaired the Board.

3.10.8.2 IGES/PDES Organization (IPO)

The IGES Working Committee was established in early 1980 to extend and repair Version 1.0 of IGES. A Steering Committee was established at the same time to manage the operation. Together they were known as the IGES Organization. Today, the IGES/PDES Organization (IPO) is the ANSI-accredited U.S. national standards body for developing product data standards and technology. The IPO is developing separate but similar standards under the direction of two projects, IGES and PDES. The U.S. Technical Advisory Group (TAG) is a Standing Committee within the PDES Project and is responsible for formulating the U.S. consensus on SC4 ballots and issues, from both a technical and business perspective. NIST chaired and administered the IPO from its inception into the 1990s. NIST also chaired the U.S. TAG in its early years, and provided continual administrative support until 1997.

3.10.8.3 US PRO Association

The U.S. Product Data Association (US PRO) is a nonprofit membership organization. Established by industry in the early nineties, US PRO works for U.S. industry by providing the management functions for the IPO and its related activities, including the TAG to ISO TC184/SC4. US PRO supports the development, publication, and distribution of PDE standards in the U.S. Such services by the Association help remove barriers that inhibit the exchange of product data across U.S. industry supply chains. US PRO is founded and operated on the belief that advancing PDE technology will improve U.S. and global competitiveness dramatically. NIST is a Silver Patron of US PRO, and has a non-voting seat on its Board.

3.10.8.4 NIST Automated Manufacturing Research Facility (AMRF)

The AMRF was established in 1982 with joint funding from the U.S. Navy Mantech Program and the Department of Commerce. Its objective was to develop the standards and technologies needed to have totally automated and integrated flexible manufacturing systems. A significant effort went into the development of manufacturing part data descriptions (both information models and databases) for use in exchanging relevant information about parts during the design and manufacturing processes. NIST technical staff were able to develop a practical understanding of the requirements and implementation of a standard such as STEP. The AMRF program concluded in 1994.

3.11 COUNT DOWN TO BLAST OFF… THE INITIAL RELEASE OF ISO 10303

By 1990, political pressure to move on an Initial Release of ISO 10303 was now in the forefront. The technical consensus on the critical elements of STEP had also been almost accomplished. SC4 Resolution 68 (June, 1990), established the first edition of STEP to include the following parts[19]:

- Overview (10303-1)
- EXPRESS (10303-11)
- Physical File (10303-21)
- Conformance Testing (10303-31)
- Generic Product Data Model (10303-41)
- Shape Representation (10303-42)
- Presentation (10303-46)
- Draughting (10303-101)
- One or more Application Protocols as per Resolution # 62 (which stated edition 1 must include at least one draughting related AP, and ISO 10303-201 was recommended as the top priority).

Additional parts could be considered for the first edition; however, no additional part could be included if its inclusion would result in a schedule slippage for the mandatory parts.

3.11.1 Initial Release

SC4 further decided that STEP, in its Initial Release, should provide at a minimum the capability already offered by the several national standards. This meant that an application protocol for configuration management (ISO 10303-203) would be required and that the development of any other APs would not be allowed to interfere with the completion of this AP. This decreased sharply the likelihood that the Initial Release would contain more than one AP, since much work still had to be accomplished on ISO 10303-201.

[19] Actual published titles for these parts differed from those as listed here.

This edict caused a ripple effect, requiring the devotion of additional resources to complete the effort. At a minimum, integrated resources (IRs) supporting 10303-201 needed to be completed. This meant that not only the generic resources supporting product description were needed, but also two other kinds of resources: an application resource and a management resource (Figure 3-6). Of primary interest was the application resource. Establishing such a resource provided a solution germane to a large group of applications but not to all applications. In the case of 10303-201, the application resource was drafting.[20]

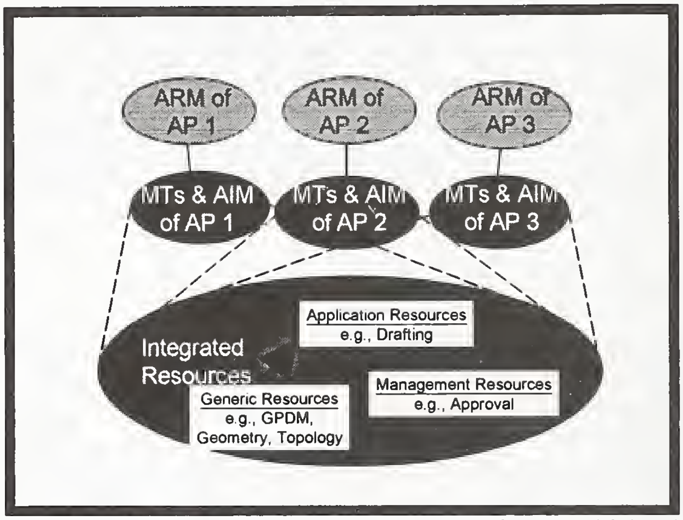


Figure 3-6: The Integration Architecture of STEP for the Initial Release

Management resources (e.g., approval) typically had different requirements in different applications. Therefore, a modular structure of independent management resources had been adopted in the IRs. At the time an AP was developed, appropriate constraints would be imported into an AIM and the data structure of the management resource completed within the AIM (e.g., what product description data needed approval in the particular application context).

Since EXPRESS did not have a specific construct to completely support the resources, a work-around was developed using several EXPRESS constructs that necessitated somewhat clumsy data structures to get the job done for the Initial Release. It was agreed that this situation would be corrected in EXPRESS in a subsequent release.

3.11.2 A Tiger Team Effort to complete the Initial Release

Although the technical elements were available, there was insufficient coordination of efforts to finish all the work to be done by the date specified by SC4. A Tiger Team was established that had the task to bring everything together. NIST played a central role in this effort, hosting many workshops and providing several of the Tiger Team's resources. The workshops were needed to review and ensure that all of the necessary elements were completed and of sufficient quality to be approved as an international standard.

Early in 1993, the Initial Release of STEP was sent out for ballot. There was a collective sigh of relief heard around the world. The ballot was successful! In December 1994 the Initial Release of STEP became an international standard – ISO 10303:1994, *Industrial Automation Systems and Integration – Product data representation and exchange,* ten years after the first ISO TC 184/SC4 meeting at NIST.

[20] This resource was eventually published as ISO 10303-101: 1994 Industrial automation systems and integration -- Product data representation and exchange -- Part 101: Integrated application resources: Draughting

3.12 NO TIME TO REST AFTER INTIAL RELEASE

Two major, unresolved issues led to another reorganization of SC4 working groups after the Initial Release of ISO 10303. The first issue was that many technical experts believed that the STEP architecture, as it existed in the Initial Release, did support data exchange but did not support data sharing. They believed that to support data sharing, a context-independent approach was needed. (A minority opinion held that data sharing was enabled by the existing STEP architecture and methods.)

The second issue was induced by a procedural requirement on APs called "interpretation." The process of interpretation is described in Chapter 4. Its mention here serves only to highlight a resource issue. There were so few experts with an in-depth knowledge of the IRs, the application domain of the AP, and with doing interpretation. Consequently, AP development nearly always hit a bottleneck when interpretation by SC4/WG4 was needed. Unfortunately, interpretation was also necessary to develop the mapping tables (MTs) and AIM, thus completing the AP. There was a general movement that the responsibility for interpretation should be with the AP development teams and reliance on the experts of WG4 should be minimized. (Of course, some domain experts likened this direction to severing the heart from the arteries, but telling the arteries they still had to pump blood!) This direction of thinking brought SC4 to disband WG4, and replace it with a Quality Committee and an additional working group, which was responsible for the integrated resources.

After the Initial Release, funding changed dramatically for many participants in SC4 working groups and the focus of their activities also changed. User investment dollars continued to flow to support development of these particular application protocols that supported their own particular domain. Meanwhile, few felt it necessary to support continuing enhancements to the IRs. Fewer still appreciated providing resources to assist in the intensive internal quality and integration requirements necessary to meet SC4 edicts.

A new undercurrent across SC4 saw the architecture and methods of the STEP Initial Release as having serious flaws. This new undercurrent gained momentum and a reorganization of the STEP working groups became imminent. Again. Through a sequence of events and resolutions,[21] WG10, Technical Architecture, was created to build an architecture to support all of the SC4 standards. This immediately added several components to the complexity of the task: the developing ISO 13584 (Parts Library), and the still fledging efforts on ISO 15531 (MANDATE) and ISO 14959 Parametrics. Working Groups 4, 5, 6 and 7 were disbanded; their functions absorbed into the newly created Working Groups 11 and 12, and the Quality Committee.

3.12.1 Effect on AP Development

Though the organizational structure and practices of SC4 working groups had changed, much of the development of APs continued using what might be called Classic STEP [42] (i.e., using the architecture and methods of the Initial Release). However, other efforts were underway by technical experts to adopt new approaches both in application areas and in information modeling technology.

Both Classic STEP and a new emerging architecture and methodology were being used by different developing APs (e.g., 10303-227[43] and 10303-221 [44] respectively). The new methodology (used in 10303-221) was based on a specialization of very abstract information classes and followed suggestions made by EPISTLE [45], a consortium of principally European companies that had developed the methodology. This classification scheme though similar in approach to Classic STEP, was totally different in its details. The intent was toward a universal context for all data, thus returning to the idea of a single, integrated model (similar in structure but entirely different in detail to the IPIM).

The proposed EPISTLE class architecture and methodology used by 10303-221 presented a rather challenging interpretation problem. 10303-227 was developed under the Classic STEP approach and therefore its model was considered an ARM that needed interpreting to develop an AIM from the IRs. The IRs used generic, product-

[21] See Chapter 9 for more detail.

description data constructs. 10303-221, using the EPISTLE approach, used generic data constructs as its starting point and specialized from there to arrive at classes appropriate to their requirements. Given these circumstances, the only approach envisioned for 10303-221 was to interpret the classes as database representation requirements for the identified scope of 10303-221. This meant that these classes would be interpreted in terms of product data property representations in the IRs. This was unacceptable to developers of 10303-227 and this disagreement remains an open issue. Both these APs are part of a suite of APs for the process plant industry.

3.12.2 Effect on the STEP Architecture

WG10 has considered two approaches to the SC4 Architecture with respect to satisfying the data sharing requirement for STEP. The first is to change the Classic STEP architecture to either the EPISTLE proposal or one of several other proposals that each have a single universal context. This is the most radical approach, since STEP is already an ISO standard. Over the course of several years none of the proposals gained consensus, but many are still under consideration by WG10.

The second approach would essentially allow the development of STEP APs to use the existing STEP architecture and methodology (even though the differences of 10303-221 were considered). This approach would turn its attention to an umbrella architecture and methodology for SC4 that would include a migration path for those STEP APs that did not meet the requirements of a new architecture. This approach has not achieved broad agreement either and is still under debate.

WG10 was asked by SC4 if standards in SC4 should be authorized using other than the STEP architecture and methodology (for which there was already a precedent). WG10 passed a resolution that did not prohibit such work [46]; however, the following comments were offered:

Pros
Such standards --
1. Will produce a data exchange capability within a specified application domain.
2. May be effective in attracting industry support and vendor take-up.
3. May be used as the basis of industrial trials prior to developing an AP.

Cons
1. Creation of such standards will require considerable investment to create procedures for development, validation, testing of implementations, etc.
2. Standardization of competitive, non-integrated specifications for product data exchange will have a negative impact on the SC4 goal of consistent architectures for industrial data.
3. Creation of such standards may divert scarce human resources from ongoing SC4 standards development.

This resolution facilitated the establishment of a new standard activity in SC4 (ISO 15926) for the oil and gas industries that will not develop an AP according to the STEP architecture and methods. The new standard will be using what might be characterized as an EPISTLE-like architecture and methodology.

The idea of an SC4 architecture as an umbrella for all SC4 standards, including STEP, has not reached consensus as of the writing of this manuscript. An SC4 architecture would be required if the work of ISO 10303 and ISO 15926, which both fall within the scope of product data, are to be reconciled.

3.12.3 Interpretation Bottleneck in AP Development

AP development teams now have the major responsibility for interpretation under the current organization. This new responsibility is slow to be inherited because, in part, there are less than six experts from around the world available to perform interpretation. In addition, there continues to be some debate over the importance of integration, especially for those APs outside a particular industrial domain. If integration occurs across the APs in a particular domain, what value is added to integrate those same APs with another domain? AP developers continue to seek an approach that reduces the intensive time and associated cost to develop an AP.

Work initiated by PDES Inc. on modularization of APs, in its AP development activities, was proposed to WG10 in 1997 [47]. PDES Inc. is redefining 10303-203 using modular principles. They have suggested first to WG10 and then to SC4 that teams developing new APs consider structuring their information requirements in a modularized form. More explanation on modularization can be found in Chapter 10.

3.13 CONCLUSION

Over the course of ISO 10303 development, resolving what appeared to be irreconcilable islands of consensus was dealt with through technical and political means. Technically, countless ad hoc groups were formed to hash out an answer to an issue. Politically, the sources of funding for ongoing work and the economic agendas of these funding sources often prevailed. Sometimes a reorganization helped to prioritize the work, to prevent a different focus, or to better align the efforts.

The single, irrefutable fact that has held true throughout the development of STEP is that the struggle never ends. Cycles of building consensus are an ever-present reality, and NIST wanted to remain engaged in both the political as well as the technical struggle to facilitate adequate coordination of U.S. interests.

STEP has capabilities that span multiple industries. Those industries driving and actively developing standards today include architecture & construction, aerospace, automotive, electrical & electronic, manufacturing technologies, process plant, and shipbuilding. STEP standard parts, implementation software, and methods are being deployed in other standards development efforts. Many of the functional advantages associated with STEP are highlighted in the following chapters, but in summary include:

- Context-based communication.
- A multi-application systems focus.
- Product type focus.
- Lifecycle and industrial use focus.
- Non-proprietary focus.
- Standardized conformance-testing criteria.

The top ten CAD vendors have built, or have committed to building, STEP translators for the initial release of 10303-203. Several product data management (PDM) vendors have made similar commitments. Even as the Standard Data Access Interface (SDAI) [48] becomes an international standard, several vendors are committed already to providing support products in one-to-many of the language bindings (ISO 10303 series of parts 23-26) to allow interface.

The real message in successful deployment of STEP lies with the user. The following are excerpts from news releases declaring corporate commitments to ISO 10303:

> **LOCKHEED MARTIN IMPLEMENTS NEW DATA EXCHANGE STANDARD**
> As part of its ongoing effort to foster product affordability, Lockheed Martin Tactical Aircraft Systems (LMTAS), in Fort Worth, Texas, recently implemented an international data exchange standard known as the Standard for the Exchange of Product Model Data, or STEP, on its F-16, F-22, Joint Strike Fighter, F-2 and KTX-2 programs... [49].
>
> **BOEING IMPLEMENTS STEP AS PRODUCTION EXCHANGE PROCESS WITH THREE ENGINE COMPANIES**
> Boeing Commercial Airplane Group has agreed with Pratt & Whitney, Rolls-Royce and GE Aircraft Engines to use STEP as the production data exchange process in support of the 777 and 767-400 Extended Range programs. STEP will also be the preferred process for future programs.

Boeing and the engine companies exchange product data in support of the Digital Pre-Assembly (DPA) process, which verifies the form and fit of the parts that integrate the airplane engine and the airplane. In the previous process, large assemblies of solid models were exchanged using the proprietary data format of Dassault Systemes' CATIA. The engine companies used custom-built software translators to exchange data between their CAD system and CATIA, and typically involved manual rework of the models to carry out the exchange process. Custom translators are expensive to develop and maintain [50].

GENERAL MOTORS IS THE FIRST AUTOMOTIVE COMPANY TO PUT NEW ISO DATA EXCHANGE STANDARD -- STEP -- INTO PRODUCTION
General Motors announced today that it has started production operations at its new STEP Translation Center (STC) in Troy, Michigan. The center uses the new International Standard - STEP, ISO 10303, the STandard for the Exchange of Product Model Data, to transfer product designs between teams different computer-aided design (CAD) systems. STEP replaces less effective methods of data exchange that have been barriers to streamlining the process of developing new products. The EDS-operated center is used to exchange designs new products among GM divisions, their customers and suppliers. The center will then be used to increase cooperation on the design of new products and move them into production in less time and at reduced cost...[51]

STEP GOES PRODUCTION!!!
December 13, 1995: After a year of hard work and determination, McDonnell Douglas has taken STEP into production! The PDES, Inc. supported CSTAR effort (C-17 STEP Transfer and Retrieval) successfully exchanged C-17 design data this week between McDonnell Douglas' Long Beach and St. Louis sites using ISO 10303-203 as the neutral exchange mechanism. The data transferred referred to information on the Inboard and Outboard Pylons for the C-17. Approximately 525 drawings and 2,200 parts were transferred totalling over 75 megabytes of data... [52].

Perhaps more telling are the several industrial sector commitments already on the books. Each of these industrial sector commitments is independent of national and regional boundaries.

Memorandum of Common Understanding and Cooperation on the Use of STEP (ISO 10303). This memorandum is signed by eleven major aerospace companies and is a significant milestone in representing the commitment of the participants to use STEP. Signed in October 1995, signature companies include: General Electric, Boeing, Rolls Royce Aerospace Group, Pratt and Whitney, Lockheed Martin Tactical Aircraft Systems, Northrop Grumman, Lucas, McDonnel Douglas, SNECMA, Allison Engine Company, and Hughes Aircraft. GE, Boeing, and Rolls Royce signed an earlier initial version of this agreement in December, 1994.

STEP Automotive Special Interest Group (SASIG). On December 5, 1995, SASIG wrote a letter to All Directors of CAD/CAM product development, recommending a list of implementation priorities for specific STEP application protocols and their conformance classes. The list was intended to help each company facilitate its product development and delivery schedules to meet the automotive industry's deployment of STEP. Of significant note of SASIG's message is the members of SASIG itself. Four automotive associations from four different countries comprised SASIG: Automotive Industry Action Group (AIAG, United States), Groupement pour l'Amerioration des Liasons dans I'Indstrie Automobile (GALIA, France), Japan Automobile Manufacturers Association, Inc. (JAMA, Japan), and Verband der Automobilindustrie (VDA, Germany).

These testaments of current or planned use need no further explanation other than to say commitment to developing and deploying ISO 10303 is alive and well!

CHAPTER 4

THE BUILDING BLOCKS OF STEP

4.1 CHALLENGES FOR STEP

Each part of ISO 10303 contains the following introductory paragraph that summarizes the significant challenges undertaken in this standardization effort:

> "ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the lifecycle of a product, independent from any particular system. The nature of this description makes STEP suitable not only for neutral file exchange, but also as a basis for implementing, sharing product databases, and archiving [53]."

The following list provides greater detail to the initial objective for STEP mentioned in Chapter 3. It is compiled from a number of sources, including the draft "Architecture and development methodology reference manual" [54]:

1) The scope of STEP includes all product data for any stage of the product lifecycle for any industry.
2) STEP shall support the complete and unambiguous exchange of product data between application systems.
3) STEP shall support the complete and unambiguous archiving of product data. STEP shall enable the lifetime availability of data.
4) STEP shall support the sharing of product data between application systems.
5) STEP shall provide improved reliability and efficiency from other standards.
6) STEP shall support upward and downward compatibility of implementations. STEP shall be extensible and must support change.
7) Compatibility with other standards is a requirement for STEP.
8) Implementations of STEP shall be testable to facilitate user acceptance of the standard.

STEP was designed to be the successor of such exchange standards as IGES, SET, and VDA-FS (discussed in Chapter 2) with the notable difference that it was intended to do more than support exchange of product data. STEP is intended to support data sharing and data archiving. These distinguishing concepts are described in an SC4/WG10 document [55] in the context of the STEP architecture, and are paraphrased below:

Product data exchange: the transfer of product data between a pair of applications. STEP defines the form of the product data that is to be transferred between a pair of applications. Each application holds its own copy of the product data in its own preferred form. The data conforming to STEP is transitory and defined only for the purposes of exchange.

Product data sharing: the access of and operation on a single copy of the same product data by more than one application, potentially simultaneously. STEP is designed to support the interfaces between the single copy of the product data and the applications that share it. The applications do not hold the data in their own preferred forms. The architectural elements of STEP may be used to support the realization of the shared product data itself. The product data of prime interest in this case is the integrated product data and not the portions that are used by the particular product data applications.

Product data archiving: the storage of product data, usually long term. STEP is suitable to support the interface to the archive. As in product data sharing, the architectural elements of STEP may be used to support the development of the archived product data itself. Archiving requires that the data conforming to STEP for exchange purposes is kept for use at some other time. This subsequent use may be through either product data exchange or product data sharing.

Early in the development of ISO 10303, SC4 recognized that the scope of the standard was extremely large. The Complainers and Gripers Ad Hoc Committee noted this as an issue in its 1987 report [56]. This fact resulted in a couple of fundamental assumptions that shaped the architecture of STEP. SC4 assumed it unlikely that any one organization would implement the entire ISO 10303, due to its large scope. Therefore, it made sense to separate the standard into parts, where an organization would implement only the subset of parts needed to satisfy the requirements of their operation. Secondly, SC4 assumed that the appropriate way to subdivide the large scope of STEP into parts was by views of product data; meaningful exchanges of product data happen only when the applications share a common context.

Another primary concept contributing to the architecture is that the content of the standard is to be completely driven by industrial requirements. This, in combination with the concept that the re-use of data specifications is the basis for standards, led to developing two distinct types of data specifications. The first type, reusable, context-independent specifications, are the building blocks of the standard. The second type, application-context-dependent specifications (application protocols) are developed to satisfy clearly defined industrial information requirements. This combination enables avoiding unnecessary duplication of data specifications between application protocols.

SC4 determined that computer-sensible standards specifications were necessary to facilitate reliability and efficiency. The expression of STEP data constructs through a formal data definition language is necessary (but not sufficient) for unambiguous definition of data.

SC4 also determined it necessary to separate the data definition from the exchange format and the data access language to best facilitate data exchange, data sharing, and data archiving. Separating data specifications from the method of implementation has two advantages: the data specifications may be extended without requiring changes to the implementation method and a single data representation may be used with each implementation method.

A lesson learned from ISO 9646 Open Systems Interconnection (OSI) standards [57] was the need to incorporate a built-in basis for assessing conformance of implementations into the STEP architecture. SC4 made the decision to go one step further than OSI and standardize abstract test suites as well as the testing methodology and framework. Standardized abstract test suites are a prerequisite to repeatability and consistency of testing, and therefore promote recognition of test results across test laboratories. Chapter 8 provides more detail on the merits of the architectural components for conformance testing.

4.2 COMPONENTS OF ISO 10303

The architecture of STEP is intended to support the development of standards for product data exchange and product data sharing. (Some debate was mentioned in the previous chapter over whether this support is adequate.) The requirements and concepts in the preceding section have contributed to the evolution of the architecture over the past decade. The architectural components of STEP are reflected in the decomposition of the standard into several series of parts. The STEP document composition was developed at the June 1989 meeting of ISO TC184/SC4/WG1 as a series of parts. Each part series contains one or more types of ISO 10303 parts. Figure 4-1 provides an overview of the structure of the STEP documentation.

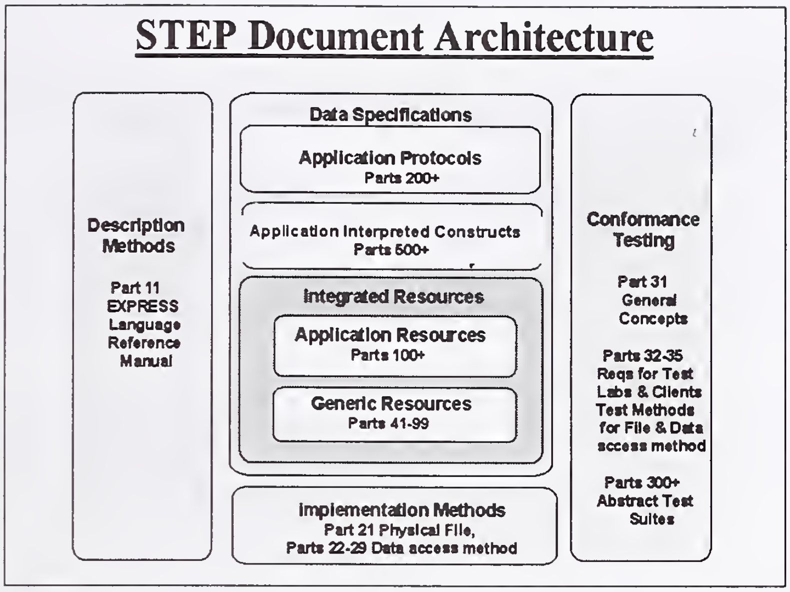


Figure 4-1: Overview of the STEP Documentation Structure

The following describes each of the structural components and functional aspects as an overview of the STEP architecture.

4.2.1 Description Methods

The first major architectural component is the description method series of STEP parts. Description methods are common mechanisms for specifying the data constructs of STEP. Description methods include the formal data specification language developed for STEP, known as EXPRESS [58]. Other description methods include a graphical form of EXPRESS, a form for instantiating EXPRESS models, and a mapping language for EXPRESS. The Description methods are standardized in the ISO 10303-10 series of parts. Various aspects and use of the EXPRESS language are described in further detail in Chapter 5.

Developing a data specification language specific to STEP is an innovative approach to standards development. Existing languages were evaluated for use within STEP, but none satisfied all the requirements of the standard. The fact that the EXPRESS language is used outside of ISO 10303 and even outside of the ISO subcommittee in which it was developed is evidence of its utility. The perhaps somewhat painful history of EXPRESS adoption was described in Chapter 3.

4.2.2 Implementation Methods

The second major architectural component of STEP is the implementation method series of 10303 parts. Implementation methods are standard implementation techniques for the information structures specified by the only STEP data specifications intended for implementation, application protocols. Each STEP implementation method defines the way in which the data constructs specified using STEP description methods are mapped to that implementation method. This series includes the physical file exchange structure [59], the standard data access interface [60], and its language bindings [61,62,63]. Implementation methods are standardized in the ISO 10303-20 series of parts. Chapter 6 discusses implementation methods in further detail.

Separating the data specification from the implementation method is another SC4 creation found in STEP. This separation enables upward and downward compatibility of implementations of STEP – in theory. In practice, there are a whole host of other issues that impact upward and downward compatibility. One of the strongest technical and political forces in resolving these issues is via the vendor and the associated impact on the vendor's implementation.

4.2.3 Conformance Testing

The third major architectural component of STEP is in support of conformance testing. Conformance testing is covered by two series of 10303 parts: conformance testing methodology and framework, and abstract test suites. Chapter 8 discusses conformance testing in further detail.

The conformance testing methodology and framework series of 10303 parts provide an explicit framework for conformance and other types of testing as an integral part of the standard. This methodology describes how testing of implementations of various STEP parts is accomplished. The fact that the framework and methodology for conformance testing is standardized reflects the importance of testing and testability within STEP. Conformance testing methods are standardized in the ISO 10303-30 series of parts.

An abstract test suite contains the set of abstract test cases necessary for conformance testing of an implementation of a STEP application protocol. Each abstract test case specifies input data to be provided to the implementation under test, along with information on how to assess the capabilities of the implementation. Abstract test suites enable the development of good processors and encourage expectations of trouble-free exchange.

NIST, working with representatives from the United Kingdom and France, was instrumental in establishing the requirement that abstract test suites be standardized in ISO 10303. Several of the STEP conformance testing concepts were modeled after the ISO 9646 [64] series of parts. This standard helped establish a foundation for concepts, methods, and vocabulary. SC4 also had the advantage of learning from the ISO 9646 mistakes. By not standardizing the ATSs in the OSI example, one was never assured an ATS would exist for testing an implementation against a particular OSI application. No assurance of an ATS meant no assurance for an ability to test an implementation of the standard. SC4 hoped by standardizing ATSs it would:

- Bring appreciation to the forefront for the requirement.
- Ensure resources were available to carry out the preparation of the ATS.
- Make available the ATS as the AP was being finalized.
- While under development, use the ATS to reaffirm or correct the developing AP.
- Keep consistent the methodology and concepts across ATSs.

Five years after SC4 Resolution 168,[22] SC4 has many lessons to share with the next standards developers planning to standardize ATSs. Abstract test suites are standardized in the ISO 10303-300 series of parts, although now they are first produced as Type II Technical Reports. Such reports are an intermediate stage to finalizing a standard to allow complex, unstable ideas to be tested and implemented before proceeding to maturity. The SC4 community has been slow in learning what is necessary to build an ATS to support a 10303 AP. NIST provides some of the few experts in the world for this technical development.

4.2.4 Data Specifications

The final major component of the STEP architecture is the data specifications. There are four part series of data specifications in the STEP documentation structure, though conceptually there are three primary types of data specifications: integrated resources, application protocols, and application interpreted constructs. All of the data specifications are documented using the description methods.

---

[22] ISO TC 184/SC4 Resolution 168 stated: An Abstract Test Suite shall be developed to document the guidelines for testing implementations for conformance to the Application Protocol. Before an Application Protocol can be registered as a DIS, the corresponding Abstract Test Suite must at least have started its CD ballot.

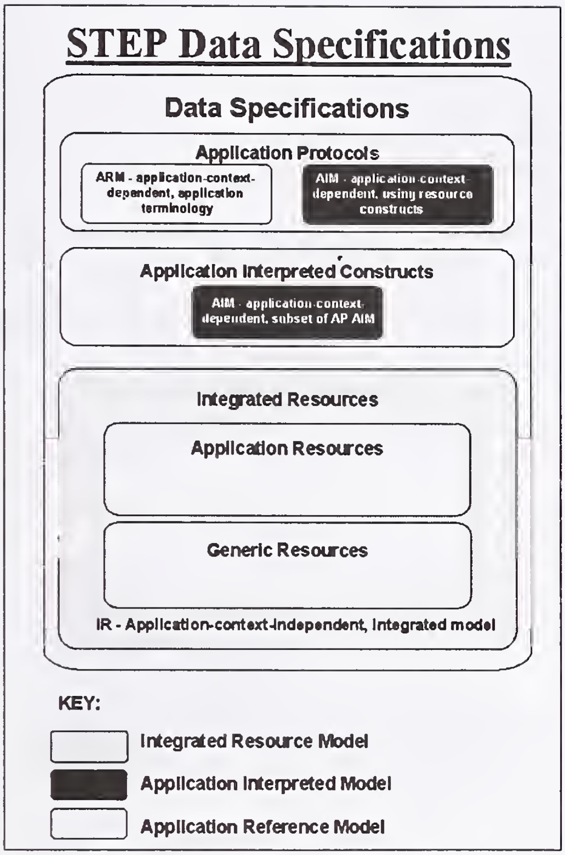


Figure 4-2: STEP Data Specification

4.2.4.1 Integrated Resources

The integrated resources constitute a single, conceptual model for product data. The constructs within the integrated resources are the basic semantic elements used for the description of any product at any stage of the product lifecycle. Although the integrated resources are used as the basis for developing application protocols, they are not intended for direct implementation. They define reusable components intended to be combined and refined to meet a specific need.

The integrated resources comprise two series of parts, the integrated generic resources and the integrated application resources. The two series have similar function and form: they are the application, context-independent standard data specifications that support the consistent development of application protocols across many application contexts. These are data models that reflect and support the common requirements of many different product data application areas.

Examples of generic resource constructs include Cartesian point, date, and product. These constructs could potentially be used by any application. Integrated generic resources are standardized in the ISO 10303-40 series of parts. The current integrated generic resources cover:

- Product description and support (ISO 10303-41).
- Geometric and topological representation (ISO 10303-42).

William Burkett recalls…On June 11th, 1986, the "Peoria Project" kicked off. This was a concentrated, focused effort that eventually produced the Product Structure/ Configuration Management (PSCM) model. This model evolved into Part 44. Three full-time participants (Tom Voegeli, Caterpillar; Ravi Krishnaswami, GM/EDS; and Mike Yinger, Northrop) spent the summer in Peoria, Illinois to develop this piece of work. As a part-time participant, I traveled to Peoria from St. Louis every two weeks or so - Ravi later said it was the best summer of his career!

- Representation structures (ISO 10303-43).
- Product structure configuration (ISO 10303-44).
- Materials (ISO 10303-45).
- Visual presentation (ISO 10303-46).
- Shape variation tolerances (ISO 10303-47).
- Process structure and properties (ISO 10303-49).

NIST has repeatedly played a critical technical role in developing integrated resources. Also, to better leverage or integrate STEP development with other existing standards, NIST often served as the technical conscience to SC4. In the late 1980's, the visual presentation resource was being developed "fresh and new," and driven primarily by interests in Germany. NIST believed there were functional capabilities that could be leveraged from ISO 9592, Programmer's Hierarchical Interactive Graphics System (PHIGS) [65].

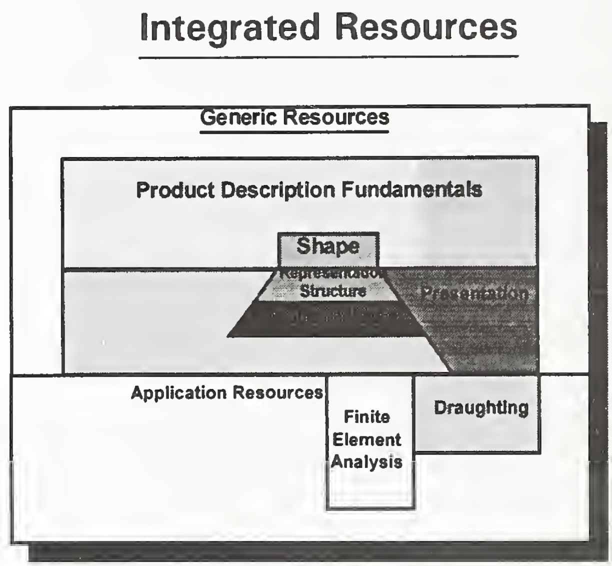


Figure 4-3: Integrated Resources

NIST sponsored the ISO/IEC JTC1 PHIGS Working Group Chair (who just so happened to be from the U.S.), to participate actively in the continued development of the presentation model in SC4. Although the NIST funding for this push was short-lived, the lingering results of this effort can be seen today in ISO 10303-46 where PHIGS and elements of the ISO Graphic Kernel System (GKS)[66] are referenced informatively. NIST also led the effort to produce another integrated resource, ISO 10303-45[67], as well as the earlier developmental efforts of ISO 10303-47 [68].

Integrated application resources represent concepts related to a particular application context that supports common requirements of many other product data applications. Examples of application resource constructs include drawing sheet revision, drawing revision, and dimension callout. These constructs may be used by any application that includes drawings. Integrated application resources are standardized in the ISO 10303-100 series of parts. NIST participated in developing ISO 10303-101.

They provide reusable information and a consistent foundation for STEP application protocols; however, one of the biggest challenges SC4 has faced is trying to build integrated resources in parallel with the standards needing those resources. It often raises debate over content, timing for release, and adequacy of coverage offered by these resources.

4.2.4.2 Application Protocols

Application protocols (APs) are the implementable data specifications of STEP. APs include an EXPRESS information model that satisfies the specific product data needs of a given application context. APs may be implemented using one or more of the implementation methods. They are the central component of the STEP architecture, and the STEP architecture is designed primarily to support and facilitate developing APs.

The elements of an application protocol are shown in Figure 4-4.

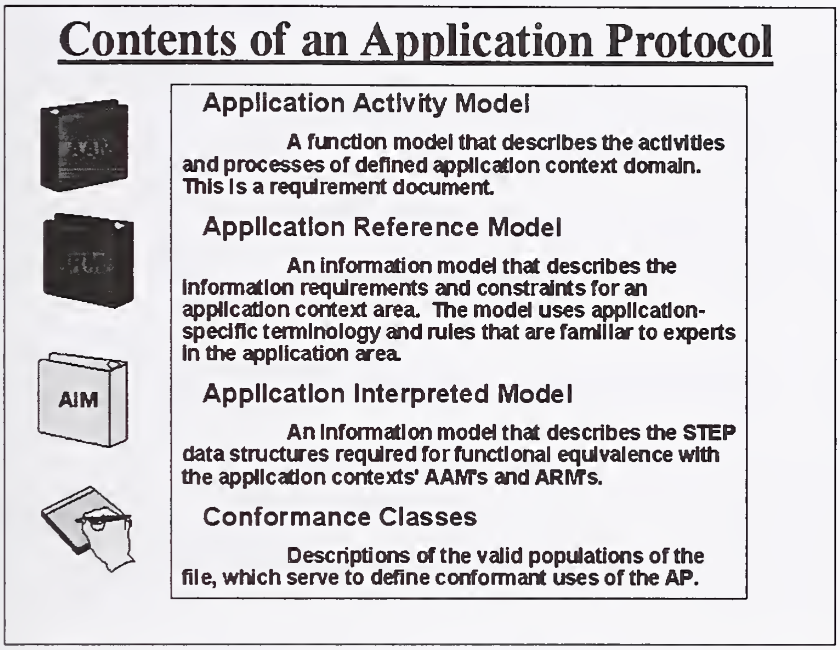


Figure 4-4: Contents of an Application Protocol

Many of the components of an application protocol are intended to document the application domain in application-specific terminology. This facilitates the review of the application protocol by domain experts. The application interpreted model (AIM) is the component of the AP that is the normative, implementable information model in EXPRESS. Conformance classes are defined subsets of the AIM that may be used as a basis for conformance testing of implementations. APs were introduced conceptually as early as IGES and their evolution is covered in Chapters 2 and 3. Chapter 7 discusses in detail APs today. Application protocols are standardized in the ISO 10303-200 series of parts.

4.2.4.3 Application Interpreted Constructs

Application interpreted constructs (AICs) are data specifications that satisfy a specific product data need that arises in more than one application context. An application interpreted construct specifies the data structures and semantics that are used to exchange product data common to two or more application protocols. Application protocols with similar information requirements are compared semantically to determine functional equivalence that, if present, leads to specifying that functional equivalence within a standardized AIC. This AIC would then be used by both application protocols and available for future APs to use as well. STEP has a requirement for interoperability between processors that share common information requirements. A necessary condition for satisfying this requirement is a common data specification. Application interpreted constructs provide this capability. Application interpreted constructs are standardized in the ISO 10303-500 series of parts.

Conceptually, the purpose of AICs is sound. Once a library of AICs exists from which developing APs can draw, AP development time and an AP specification's physical size should be reduced. Nevertheless, SC4 ran into a similar development timing and configuration problem as that faced with the integrated resources. The scope of an AIC is based on the content of an existing ISO standard AP. Reintroducing a portion of that standardized AP into the ISO consensus-building process leads to potential changes in content. The result may be a standardized AIC that differs from the originating standard. Hence, a configuration problem exists between two ISO 10303 standards. To combat this, SC4 has initiated the concept of modularization. Mentioned briefly in the previous chapter, Chapter 10 covers this initiative in more detail.

4.3 STEP METHODOLOGY

The STEP methodology supports developing APs and the resources required by those APs. A principal feature of the STEP architecture is the layering of data specifications. Of primary interest are the context-independant integrated resources and the context-dependant application protocols. There are three classes of information models specified within these two types of specifications. The first class of information model is a collection of standardized EXPRESS schemas that are contained in the integrated resources. Each integrated resource schema is a representation of a specific subject area within the domain of product data. The integrated resources are abstract, conceptual structures of information that are generic with respect to various types of products and different stages of the product lifecycle. The process of ensuring that STEP integrated resources form a cohesive whole is called resource integration.

The second and third classes of information models are contained in application protocols: the application reference model (ARM) and the application interpreted model (AIM). An ARM captures the information requirements for an application context that has a scope bounded by a specific set of product types and product-lifecycle stages. ARMs are presented informatively in one of two graphical modeling languages (IDEF1X or EXPRESS-G) as well as normatively in text. An AIM is an EXPRESS schema that selects the applicable constructs from the integrated resources as baseline conceptual elements. An AIM may augment the baseline constructs with additional constraints and relationships specified by entities containing local rules, refined data types, global rules, and specialized textual definitions. Today an AIM is documented in three forms within an AP:

1. An EXPRESS short listing containing inter-schema references and constructs unique to the specific subject area.
2. A completely independent EXPRESS expanded listing where all references are resolved.
3. A set of EXPRESS-G diagrams that correspond to the expanded listing.

The process of identifying the baseline constructs and specifying the additional constraints and relationships necessary to satisfy the information requirements defined in the application reference model is called application interpretation.

There are several documents that describe in detail the development methods for the data specification of STEP [69, 70, 71, 72]. The focus of this section is on the two primary principles of the STEP methodology: resource integration and application interpretation. They are the basis for developing the different types of STEP data specifications. The same pitfalls with these requirements were already mentioned in Chapter 3; hopefully the following text will build a better appreciation for these requirements.

4.3.1 Resource Integration

Resource integration brings together like elements -- information models. The result of the STEP integration process is a single information model, documented in multiple schemas in multiple standards.

Resource integration requires the application of both semantic and syntactic integration rules against draft integrated resource models. The principles governing the development of the semantic and syntactic rules, as well as the rules themselves, were documented by Danner, Sanford and Yang [73]. Below are the ten principles used to develop the rules for resource integration:

1. STEP must contain a cohesive and functionally adequate integrated resource for application protocol interpretation that has an architecture that reduces the impact of change in a phased release environment. It is important to produce a successful STEP Version 1.0 with the ability to add and modify constructs for future versions.
2. STEP will be a collection of parts, each of which is an individual standard with its own scope and unique content. The content of parts containing semantic constructs is to be conceptual in nature.
3. Constructs are to be within the scope of product data.
4. Constructs are provided for supporting application requirements.
5. Constructs are to be functionally adequate for the stated purpose.
6. Constructs are to be functionally unique (i.e., non-redundant).
7. Constructs are to be stable, complete, and correct.
8. Constructs can build upon (i.e., specialize) the semantics of other, more generic constructs.
9. Constructs included in a version of STEP are to have an explicit place and role within the schema architecture of the STEP Integration Framework [74].
10. Constructs included in a version of STEP are to be thoroughly integrated with one another.

During the process of resource integration, draft resource models are analyzed to determine their underlying meaning. The concepts represented by the draft models are compared with the concepts represented in the integrated resources. Integrated resource constructs are evaluated in terms of conceptual uniqueness and functional adequacy. Each integrated resource construct is established to fulfill a particular application context requirement. Draft resource models are harmonized to ensure conflicts are resolved, redundant constructs are eliminated, and modeling is consistent.

The placement of the draft constructs with respect to the STEP data architecture is determined. The STEP data architecture is the structure of the conceptual model. During integration, the draft constructs are aligned structurally with the existing integrated resources. This allows voids to be identified and resolved. Structuring provides completeness, structural consistency, and structural precision with respect to semantic intent.

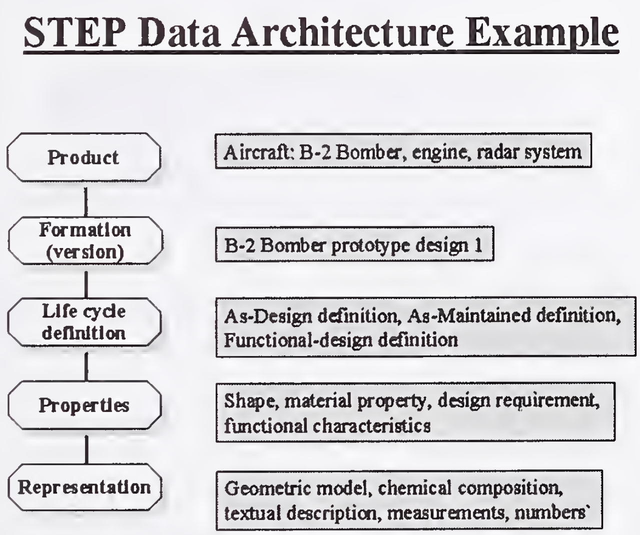


Figure 4-5: STEP Data Architecture

Conceptual and structural relationships to other constructs in the integrated resources are also defined. References between constructs are controlled to ensure consistency and manageability of the specialization. References between concepts that are in different EXPRESS schemas follow the principle of existence dependence: the more specialized constructs reference the more general constructs. Controlled references minimize the impact of change and maximize upward compatibility.

4.3.2 Application Interpretation

Application interpretation brings together unlike elements -- the information requirements of an application context and an information model. The result of the interpretation process is a single information model – an AIM.

Application interpretation is the process by which meaning is assigned to an abstract representation of an event, object, or concept. Within an application interpreted model, the abstract representation is specified by an EXPRESS construct. Interpreting an integrated resource construct results in the creation of a new construct in the application interpreted model that may restrict, narrow, or constrain the semantic scope of the integrated resource construct, thereby specializing the construct.

Application interpretation is grounded ultimately in human understanding. Application interpretation draws not only on the meaning of the constructs themselves, but on the context within which the constructs are generated, used, or received. Such things as the lifecycle stage and the application domain that bounds the scope of the application protocol defines the context specified in the application reference model. The scope of STEP is the representation of product data; the integrated resources were developed within the framework of that context. The context of the integrated resources is not limited to a specific lifecycle stage or to a type of product. Because the integrated resources and the application reference models are developed as representations of information in different (though established and related) contexts, they are subjected to the influence of different contextual factors. Interpreting integrated resources in STEP (particularly in selecting integrated resource constructs) relies on human comprehension of the requirements represented in the application reference model. STEP resources are developed through interpretation and the in-depth knowledge of the semantics and contextual factors.

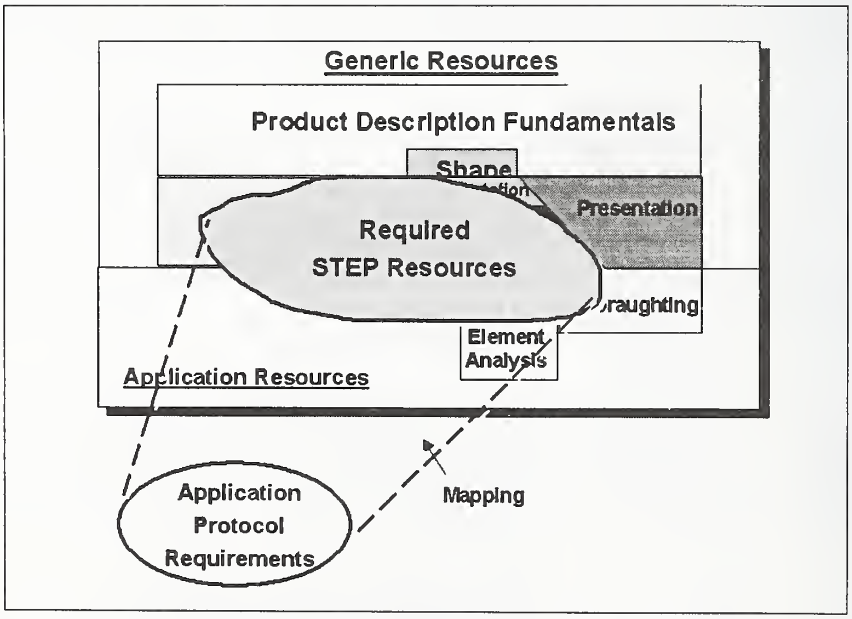


Figure 4-6: Mapping between AP requirements and required STEP resources

The STEP integration architecture requires that a minimum set of constructs be included in the application interpreted model to ensure the completeness of the semantics of product information. Regardless of its domain and scope, each application protocol must include resource constructs that identify the product, the product type, the version of the product, the product definition, and the applicable lifecycle stage of the definition.

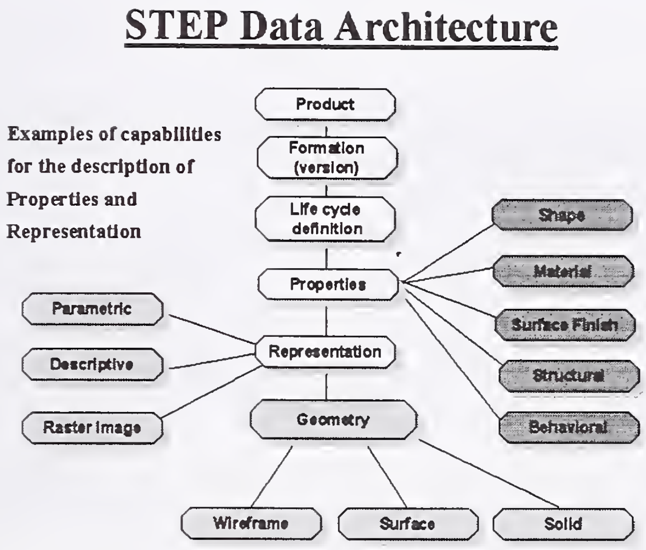


Figure 4-7: STEP Data Architecture Example

An essential part of the interpretation process is to ensure the consistent interpretation of the same requirements found in different APs. Data sharing and communication among implementations of various application protocols cannot be accomplished without consistent interpretation of the resource constructs. Consistency is achieved when the same integrated resource constructs, specializations, and constraints are specified for the same requirements in different APs.

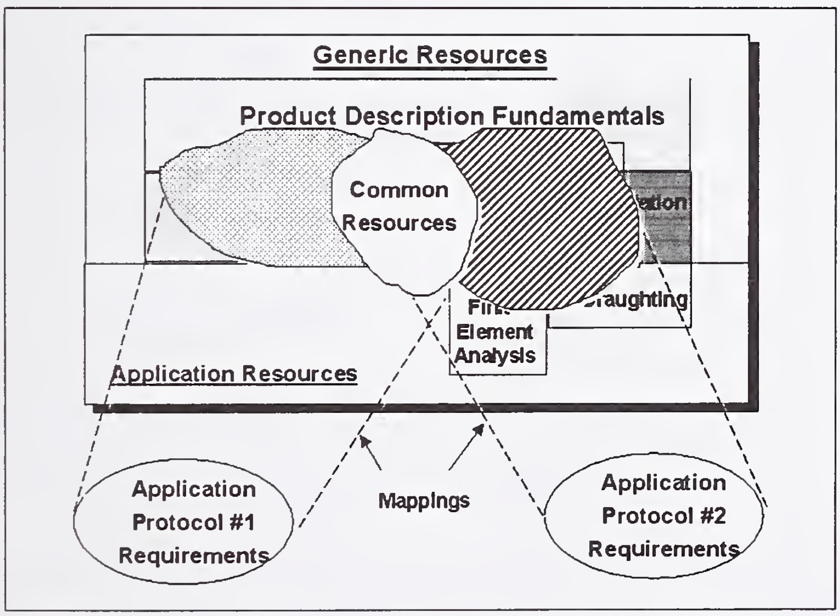


Figure 4-8: Mapping AP #1 requirements to AP #2 requirements

4.4 ARCHITECTURAL ISSUES FACING STEP

As highlighted in Chapter 3, the work of SC4 is not yet done. The following is a brief description of architectural issues facing the SC4 community following the initial release of ISO 10303.

Interoperability of applications--The ability to better communicate information among heterogeneous computer systems is still necessary.

Cooperative use of STEP application protocols-- SC4 is rich with a diverse mixture of application domain requirements from a diverse industry base. This makes cooperation with other STEP AP developers and with other standards development organizations (SDOs) more difficult.

Integrating data--Managing data from the diverse sources previously mentioned in an efficient and effective manner continues to be a challenge.

Developing data sharing implementations-- Solutions that combine STEP data models with database management system technologies to facilitate such manufacturing practices as concurrent engineering and lifecycle data management are still outstanding.

Complexity and volume of STEP--What is needed is a simplification of what is standardized. Should ISO TC 184/SC4 consider standardizing the methods only and allow APs to be developed outside the standardization process by those industrial sectors with the domain-specific requirements?

Change management of ISO 10303--As mentioned previously, how is change management to be effectively handled on those existing parts being shared among the existing ISO 10303 APs and those still under development.

4.5 CONCLUSION

The STEP architecture came into existence through a concerted effort by the SC4 community over more than a decade of thought, debate, trial and error, and consensus. An effort of this magnitude still comes at a price and lessons are learned. SC4 continues to strive for a sound STEP architecture that will support its existing ISO 10303 standard parts and allow STEP to move forward into the 21st century. Chapter 10 introduces the many directions SC4 is considering. The STEP architecture of the future – whether it remains as is or is transformed – remains an unanswered question today. Figure 4-9 provides a populated summary of how this architecture looks today, with all the many 10303 initiatives considered, underway, or already internationally adopted. This Figure has become popularly known as "STEP on a Page (SOAP)" and is maintained by Jim Nell of NIST.

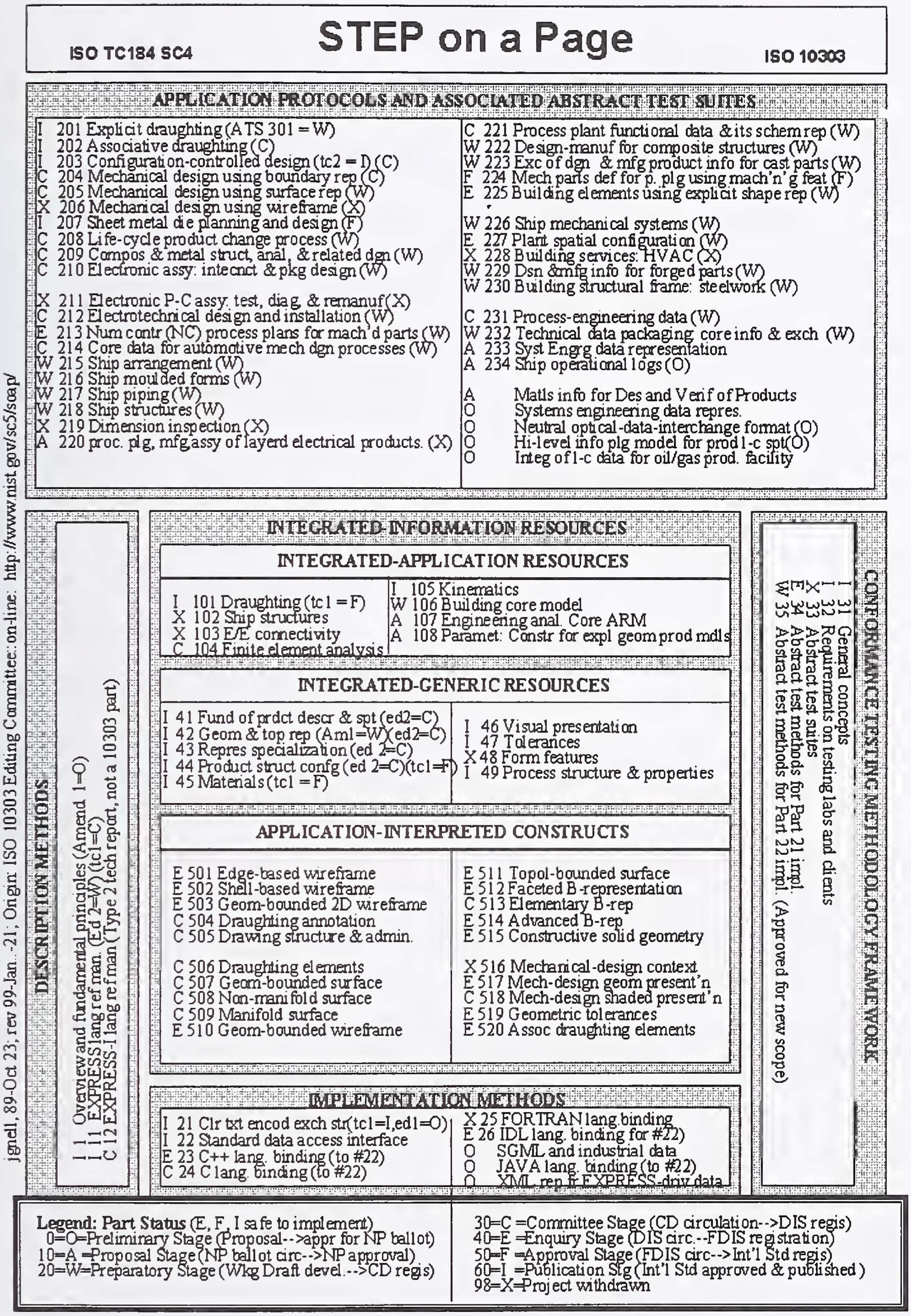


Figure 4-9: STEP on a Page

CHAPTER 5

MODELING – A WAY TO PRESENT PRODUCT DATA REQUIREMENTS

5.1 THE ROLE OF AN INFORMATION MODEL FOR DATA SHARING

ISO 10303 is a standard for product data just as a yardstick is a standard for length measurement. The yardstick itself does not tell you anything about the length of an object. Only when you place the yardstick against the object do you determine something about the object, namely its length. Likewise STEP does not tell you anything about a product but rather details the characteristics that you can use to describe the product. In order to use the standard to convey information you must apply it to a particular product. The result of that application is a set of information about, or measurements of, the product. This set of information can be encoded digitally so that software applications can process the information to perform a useful operation with respect to the product. Useful operations may be anything from displaying the version identifier of the product to conducting complex engineering analysis.

The characteristics used to describe a product are captured in an information model. The "measurements" of a product are referred to as product data. This chapter discusses information modeling in general, then discusses EXPRESS as a modeling language in more detail. STEP provides a solution to meet the fundamental requirement for manufacturing information exchange based on an agreement to develop industry data models.

5.2 ROBUST MODELING IS CRUCIAL TO STEP

As described in Chapter 3, STEP is being developed to enable complete and correct interchange of product data between various CAD/CAM systems, other manufacturing related software, and vendors [75]. A formal model is crucial to allow all parties to agree on the semantics of the information.

Factors that play a role in developing such models span the gamut from technical to non-technical. For example, technical factors include the existence and creation of robust modeling languages, techniques, protocols, and tools. Non-technical factors include the agreement on such objects (standards) and whether it is possible to buy off-the-shelf implementations. Chapter 3 discussed the history leading up to the choice of a particular modeling language for STEP.

The state of the technology impacts the technical issues while the state of the participants impacts all issues. For example, converting from one modeling system to another can be expensive enough to be unjustifiable to some companies but not to others, leaving a schism despite the existence of satisfactory technical solutions. [23]

5.3 MODELING ALTERNATIVES

Using modeling to convey information is not a new concept. The Associative Data Modeling (ADM) was conceived in 1969 by Paul Jones. Also known as the Curtis-Jones Technique, ADM provides a powerful, simple method of modeling and verifying data. ADM was made even more attractive by bringing it into the public domain [76]. The Entity-Relationship (ER) Model was proposed by Dr. Chen in 1976 [77]. It is generally considered one of the first true semantic data models to appear in the literature, although the term "semantic" was not used at the time. There are other modeling languages which contributed historically, such as the Semantic Database Model (SDM)

[23] The term "system" used here is deliberately vague and is meant to include methodologies, implementations, standards, and specifications.

published by Hammer and McLeod in 1981, which emphasized the concept of derived schema components [78]. However, these earlier modeling languages only set the stage for later modeling developments. It was these later developments such as NIAM, IDEF0, IDEF1X, and EXPRESS which are more key to STEP development. The following section provides more background and functional description on each of these modeling systems. Figures 5-1 through 5-3 are taken from an ISO paper [79] and are based on the simple manufacturing scenario:

> "The universe of discourse to be described has to do with the registration of cars and is limited to the scope of interest of the Registration Authority.
>
> Each car manufacturer has a unique name. Each car manufacturer constructs cars in several models. A car is of a particular model. A manufacturer gives a serial number to each car he produces. This serial number is unique for all cars of one manufacturer. The name of the car model is unique for all car models for all time. Any specific car model is constructed by only one manufacturer."

5.3.1 NIAM

The Nijssen Information Analysis Methodology (NIAM) is a graphical data modeling language and methodology. NIAM focuses on the analysis of natural language sentences. Each noun is represented as a node in a complex net of bi-directional, binary relationships. Each constraint either refines the relationship between two nodes or specifies a restriction among two or more relationships. The basic constructs of the NIAM language are the 'object' and the 'fact' or relationship.

NIAM has both a graphical and structured language representation, and entails an underlying methodology [80]. Today, NIAM is known as ORM – Object-Role Modeling.

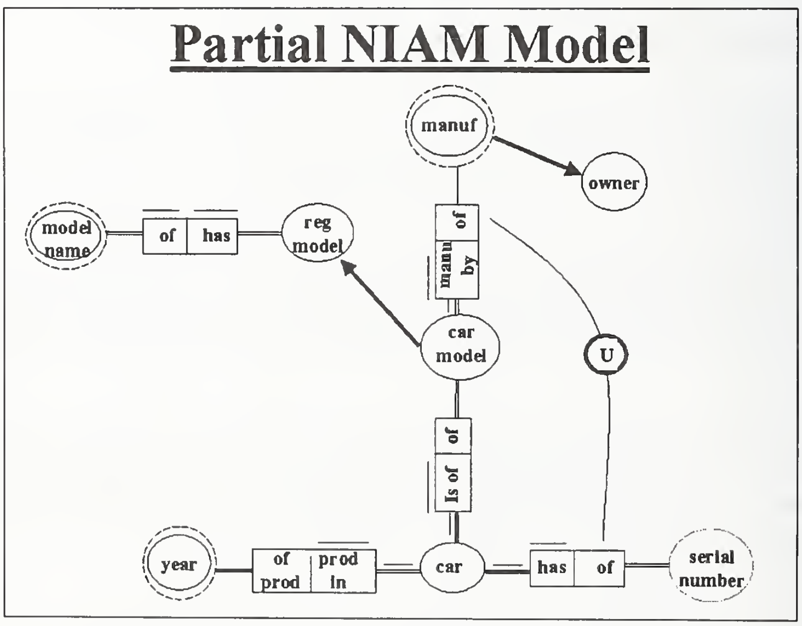


Figure 5-1: Partial NIAM Model

5.3.2 IDEF0

Chapter 2 covered the origin of the Integration Definition for Function Modeling (IDEF0) and IDEF1X. They resulted from the U.S. Air Force ICAM Program in the 1970s. IDEF0 (Integration DEFinition language 0) is based on SADT™ (Structured Analysis and Design Technique™), developed by Douglas T. Ross and SofTech, Inc. In its original form, IDEF0 includes both a definition of a graphical modeling language (syntax and semantics) and a description of a comprehensive methodology for developing models.

IDEF0 may be used to model a variety of automated and non-automated systems. For new systems, IDEF0 may be used first to define the requirements and specify the functions, and then to design an implementation that meets the requirements and performs the functions. For existing systems, IDEF0 can be used to analyze the functions the system performs and to record the mechanisms (means) by which these are done.

The result of applying IDEF0 to a system is a model that consists of a hierarchical series of diagrams, text, and a glossary cross-referenced to each other. The two primary modeling components are functions (represented on a diagram by boxes) and the data and objects that inter-relate those functions (represented by arrows).

As a function modeling language, IDEF0 has the following characteristics:

- It is comprehensive and expressive, capable of graphically representing a variety of business, manufacturing, and other types of enterprise operations to any level of detail.

- It is a coherent and simple language, providing for rigorous and precise expression, and promoting consistency of use and interpretation.

- It enhances communication between systems analysts, developers, and users through ease of learning and its emphasis on hierarchical exposition of detail.

- It is well-tested and proven, through many years of use in U.S. Air Force and other government development projects, and by private industry.

- It can be generated by a variety of computer graphics tools; numerous commercial products specifically support development and analysis of IDEF0 diagrams and models [81].

5.3.3 IDEF1X

As industry applied the modeling techniques defined by ICAM, it led to the development in 1982 of a Logical Database Design Technique (LDDT) by R. G. Brown of the Database Design Group. The technique was based on the relational model of Dr. E. F. Codd, the entity-relationship model (ER) of Dr. Chen, and the generalization-aggregation model of J. M. Smith and D. C. P. Smith. The ER model has the notion that entities, attributes, and relationships are basic semantic elements. The relational model adds rules governing well-formedness. The generalization-aggregation model contributes the distinction between generalization relationships (representing types and subtypes), and aggregation relationships (representing groupings) [82].

LDDT provided multiple levels of models and a set of graphics for representing the conceptual view of information within an enterprise. Under the technical leadership of Dr. M. E. S. Loomis of D. Appleton Company, a substantial subset of LDDT was combined with the methodology of IDEF1, and published by the ICAM program in 1985. This technique was called IDEF1 Extended or, simply, IDEF1X.

A principal objective of IDEF1X is to support integration. The IDEF1X technique was developed to meet the following requirements:

- **Support the development of conceptual schemas.** The IDEF1X syntax supports the semantic constructs necessary in developing a conceptual schema. A fully developed IDEF1X model has the desired characteristics of being consistent, extensible, and transformable.

- **Be a coherent language.** IDEF1X has a simple, clean, consistent structure with distinct semantic concepts. The syntax and semantics of IDEF1X are relatively easy for users to grasp, yet as a language, powerful and robust.

- **Be teachable.** Semantic data modeling is a new concept for many IDEF1X users. Therefore, the teachability of the language was an important consideration. The language is designed to be taught to and used by business professionals and system analysts as well as data administrators and database designers. Thus, it can serve as an effective communication tool across interdisciplinary teams.

- **Be well tested and proven.** IDEF1X is based on years of experience with predecessor techniques and had been thoroughly tested initially in both U.S. Air Force projects and private industry.

- **Be automatable.** IDEF1X diagrams can be generated by a variety of graphics packages. Commercial software is also available which supports the refinement, analysis, and configuration management of IDEF1X models [83].

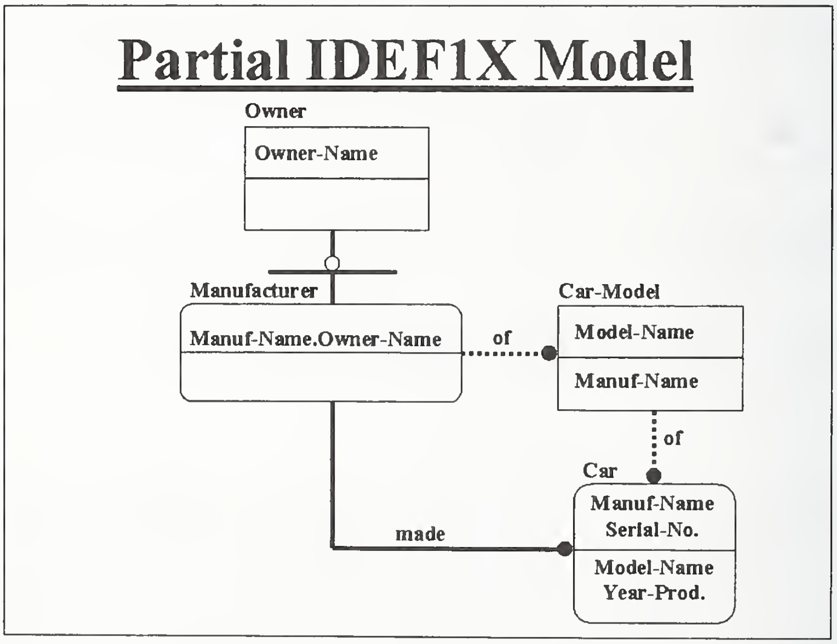


Figure 5-2: Partial IDEFIX Model

5.3.4 EXPRESS

EXPRESS [84] is designed as a language for communicating information concerning data. It has much in common with some database definition languages and some programming languages, all of which can be used to define the structure of data [85]. Unlike a database language, such as SQL [86], or a programming language, such as C [87], EXPRESS does not confuse the information modeling task with programming or database design tasks, and it is not specific to a particular programming or database system.

```
Partial EXPRESS Model

Entity owner supertype of (manufacturer);
  owner_name   :string;
unique
  owner_name;
end_entity;

entity manufacturer subtype of (owner);
end_entity;

entity car_model;
  model_name    :string;
unique
  model_name;
end_entity;

entity car;
  manufacturer           :manufacturer;
  serial_number          :number;
  model_designation      :car_model;
  model_year             :number;
unique
  serial_number, manufacturer;
end_entity;
```

Figure 5-3: Partial EXPRESS Model

5.4 MODELING INFORMATION -- WHAT'S NEEDED

STEP is unlike many other standards in information technology in that the normative form of the standard contains a computer interpretable language: EXPRESS. STEP is among the first such standards to take this approach. EXPRESS is the outcome of much debate and historical practice, review, and critique of the modeling alternatives. EXPRESS has provided distinct advantages to STEP developers and developers of STEP-capable products. These advantages are highlighted later in this chapter.

5.5 WHY DID EXPRESS COME INTO EXISTENCE?

At the time STEP started, practitioners recognized that a formal data modeling methodology was needed that divorced the issues associated with physical data structure from the semantics of the data that needed to be exchanged. This resulted in a distinction between the ISO 10303-21[88] data structure and the domain information models specified in EXPRESS. This approach was undertaken successfully in PDDI (Chapter 2), and knowledge from PDDI proved essential in introducing EXPRESS.

An early expectation of SC4 was that models would be large and must be able to be processed automatically, by well-understood techniques such as a traditional parser. Thus, IDEF1X was felt to be unsuitable because of its strictly graphical origin. IDEF1X was also disliked because it was very weak on constraint specification. NIAM was strong on constraints, but the diagrams were awkward and difficult to produce.

Therefore, when the time came to select a data modeling language for STEP, a candidate language (PDDI/EXPRESS) was already in hand that was not "owned" by someone else. The PDDI players were ready to bring their work to the SC4 table for standardization. People involved with STEP at the time also had other modeling experience with IDEF1X and NIAM. Although, as mentioned earlier in Chapter 3, NIST tried to introduce the technical merits of ASN.1[89] as a starting point for a solution, no evaluation criteria were defined and no formal evaluations of alternatives were performed by SC4. EXPRESS *would be* the answer.

A fair amount of rationalization and politics may also be blamed on the desire to invent something new for its own sake. In retrospect, was there a better choice? EXPRESS did seem to be the best choice at the time (keeping in mind that no search or evaluation was formally performed). Existing languages lacked the characteristics necessary to develop and support STEP. EXPRESS offered parsability, flexibility, ease of use by programmers, and specified constraints.

5.6 MODELING AND STEP

"Robust" modeling is important to developing STEP, but there are no quality measures or commonly accepted practices for what constitutes a resulting "good" model. "Quality" data modeling is a function of experience and since no two people have the same experiences, no two people have the same understanding of data model quality. The process of normalization with the relational database model offers some objective quality improvements, but the relational model is simple (when compared to EXPRESS) and heavily biased toward preventing data creation and update errors.

With respect to the purposes outlined above, the models in STEP play two roles. The first is to examine, explore, and understand the information within a domain. This is accomplished by the Application Reference Model (ARM). (In final form, the ARM also is used to specify the selected semantics of a domain.)

The second role of a STEP model is to specify the structure of data for the purpose of exchange. This role is played by the Application Interpreted Model (AIM). The Integrated Resources also exist for this purpose, but specify the generic semantics used in all product data domains.

5.7 CONTRIBUTIONS OF STEP TO MODELING

5.7.1 A Layered Structure

STEP has provided a large number of contributions related to modeling and the use of data models. Perhaps the most significant innovation of STEP with respect to data modeling is the layered structure used between an ARM and an AIM. This was incorporated into the architecture of the standard, as discussed in Chapter 4, and is significant because there is a specified mapping between the two. In essence, both are part of a richer, more meaningful model.

This is significant in that all data models, conceptual models, and information models are semantically "flat," i.e., "here are the data structures with the following fields and the fields mean this." There is no layering, no subroutines, no encapsulation of semantics. This is obviously an untenable situation as the data model grows to accommodate more applications or finer semantic distinctions within a single application. A single, flat schema cannot grow indefinitely --- few human minds would be able to deal with it.

The alternative is a layered approach, similar to what is done with subroutines in programs and decompositions in process models; however, while process models lend themselves to decomposition, data models do not. The closest available approach is abstraction --- the use of generalization and specialization to reduce the number of entities to a manageable number or make the semantic distinctions necessary in the domain. Thus were born the integrated resources (IRs) as a generic language for conveying information and mappings to ARMs, which refine the generic semantics to the precise (or at least more specific) semantics of the scoped domain. This data modeling innovation was introduced in STEP.

This layering may be viewed using the client-service paradigm popular in computing architectures. The IRs offer a service (generic conveyor of information) that is used by clients (the ARMs) for a specific purpose. There is no reason that this paradigm should be limited to two levels; many levels can easily be envisioned. In fact, one can treat the division between the elements of the EXPRESS language and a particular EXPRESS schema (say, the IRs) as one level, and the division between an EXPRESS schema and an ARM as another level.

5.7.2 STEP Efforts Produce Modeling Variations and Extensions

STEP efforts have also produced an abundance of modeling variations, extensions, and complements to STEP's basic use of modeling and data models. These include:

EXPRESS-G--graphical version of EXPRESS [90]. Its use is standardized in ISO 10303-11.

EXPRESS-I [91]--An instance language that provides a means of displaying example instantiations of EXPRESS-defined elements and provides formal support for the specification of test cases [92]. Because of some overlapping content with other standards, and because several of the concepts were "before their time" and needed to be validated, EXPRESS-I is published currently as a technical report.

EXPRESS-V [93]--supports two-way mappings between pairs of EXPRESS schemas, where one EXPRESS schema represents an abstract view of the other. For example, EXPRESS-V can be used to implement the mapping of entities from an Application Protocol to its ARM. EXPRESS-V is a precursor to the work on EXPRESS-X.

EXPRESS-X--Discussed in Chapter 10, supports mappings among information models defined in EXPRESS [94]. The EXPRESS-X language allows one to create alternate representations of EXPRESS models and mappings between EXPRESS models and other applications (e.g., IGES). EXPRESS-X is undergoing development within SC4 to become an international standard and part of ISO 10303.

EXPRESS-M [95]—a schema mapping language. EXPRESS-M can describe how entity instances should be mapped between schemas in order to transfer data between different models. If one follows the logic of Whitehead and Russell [96] to present-day application, EXPRESS-M was designed to plug a gap in STEP. The standard was designed to facilitate easy data sharing, but at present, there is no formal method for manipulating data during transfer. There is a major requirement for mapping entities between different APs. Two APs may describe the same product data using very different entities. To share data conformant across such APs, a mapping methodology is necessary. The concepts of EXPRESS-M are being considered as part of the developmental effort of standardizing EXPRESS-X.

ISO 10303-21 physical file exchange defines a representation for transfer of EXPRESS entity instances [97]. 10303-21 is similar to EXPRESS in many ways, in part because they were both designed in the same way by the same group of people. At the same time, 10303-21 is missing certain features that suggest exactly the opposite. For example, 10303-21 does not permit references to schemas so ambiguity exists if two schemas use the same entity name. 10303-21 also shows contradictions between whether or not it is meant to be humanly readable or editable, and whether file size is important. For example, 10303-21 does not permit the use of instance names, instead allowing only numbers as identifiers. Comment descriptors in EXPRESS are also done differently than in 10303-21.

Not surprisingly, development of 10303-21 also spanned a decade. Part of the reason for that is that continual change to EXPRESS required continual change to 10303-21. Unfamiliarity of software design practices for modern computers also contributed to the delay.

5.8 IN PRACTICE

In practice, modeling is more visible because of STEP, but has not been simplified dramatically by STEP. One could argue that while the power of the STEP modeling tools allows better models, the tools and the methodology are extraordinarily difficult to master -- to the point that very few people can perform STEP modeling expertly. Indeed, we frequently hear arguments end with an utterance that hiring a consultant is required --- since there are only a few worldwide qualified to interpret the application protocol models. This is not a good omen.

To some extent, a lack of modeling expertise in the STEP community is not surprising. Many of those expected to perform modeling are not trained in data modeling but instead are domain experts. It is difficult to find people trained in both and there is no curriculum to develop such people; however, even among data modeling experts, there is great inconsistency in and across models. These include:

- Modeling style--Existing guidelines are very superficial.
- Choosing appropriate levels of abstractions--This makes later integration very difficult.
- Modeling completeness--Virtually any model can be extended indefinitely. Everyone draws the line in a different place.
- Implementation issues--Some models account for this largely; others not at all.

While immaturity of the models is a continuing problem, an even more serious concern is maturity of the process itself. Significant problems remain in the STEP modeling arena. For example, debate continues whether different levels of models should exist and, if so, on what quantity or level of abstraction. Another unresolved issue from the initial release is EXPRESS itself, which is recognized to contain serious problems. Even if it were flawless as a modeling language, EXPRESS would not be a perfect match for STEP. Although research continues in these and other areas, STEP is under pressure to produce. As ISO 10303 matures, looking back for a complete list of lessons learned, considering the full list of alternatives, and redoing the work becomes more difficult and unlikely.

5.9 *EXPRESS* – A COMPUTER-INTERPRETABLE LANGUAGE

ISO 10303 is unlike many other standards supporting information technology. It is among the first such standards to take an approach where the normative text of the standard contains a computer interpretable language. You have already been introduced to this language -- the EXPRESS information modeling language [98]. EXPRESS has provided STEP developers and developers of STEP-capable products with distinct advantages:

(1) **EXPRESS Eliminates Ambiguity.** EXPRESS can be used to communicate among people. The EXPRESS language eliminates some ambiguity that is inevitable in natural language communication. In this way, EXPRESS sets out to do for STEP standards what Principia Mathematic[99] hoped to do for philosophy. Were ISO 10303 standard parts written solely in a natural language, erroneous interpretations would be more prevalent. Inevitably, these erroneous interpretations would find their way into STEP-capable products.

(2) **EXPRESS Assists in Validating Information Models.** EXPRESS can be used as a foundation to generate software tools that validate STEP information models. This takes the idea above one step further. Excluding a small set of 'informal propositions,' the semantics of a STEP application protocol is encoded in the AP's EXPRESS information models. Existing tools are able to process an EXPRESS information model and determine whether datasets sent to the tool conform to the information model. These tools are of great benefit in identifying errors in data generated by STEP-capable products. This technique also is functional to validate data before committing it to a database.

(3) **STEP-Capable Software Generated from EXPRESS.** EXPRESS code can be used to generate automatically high-quality STEP-capable software [100] [101] for a wide range of systems. For example, software exists that generates C++ [102] classes and methods from EXPRESS. These C++ classes implement objects corresponding to the EXPRESS entities in the information model. Generated methods may be applied to access an SQL[103] database or exchange file to instantiate objects in a client program's address space. The ability to automatically generate pre- and post-processors of the exchange format greatly reduces the opportunities for errors in the software.

(4) **EXPRESS Supports STEP Architecture.** The EXPRESS language supports aspects of the STEP architecture. The EXPRESS language provides a mechanism to refer to existing information models in the context of the current model. In terms of the STEP architecture, these mechanisms (namely the USE and REFERENCE statements)

provide the information modeler with the ability to reference and interface the STEP integrated resources as foundational notions.

(5) **EXPRESS Considered User-Friendly.** People find it relatively easy to read EXPRESS in its graphical form. The primary purpose of an information model is to make it easier for people to understand information. Graphical notations are an excellent way to convey the "big picture" of how information is organized. The graphical form of EXPRESS is called EXPRESS-G.

Generalizing on the above five points, EXPRESS benefits the information modeling and information exchange software development processes. It is not an exaggeration to say the goals of STEP could only be achieved with an information modeling language. The preceding chapter covered the benefits of EXPRESS as a modeling language. However, far from being a panacea, there are limits to what benefit any information modeling language can provide. EXPRESS falls short of the ideal and its limitations are considered later in this chapter.

5.10 EXPRESS AND VALIDATION

EXPRESS can be used to validate the correctness of the message structure between systems, which, in turn, is necessary for the successful communication of information. Data exchanged can be analyzed through use of the corresponding EXPRESS information model to determine whether it violates constraints explicitly defined in the EXPRESS. Today, validation of this sort is typically performed during system testing, not during exchange. It is reasonable to assume this sort of validation can be applied before a database transaction is allowed to commit the update to the database. To exemplify both sloppy and industrial-strength EXPRESS, perhaps a little tutorial is necessary first. Figures 5-4 through 5-7 introduce the EXPRESS terms entity, attribute, entity instance, and entity data type.

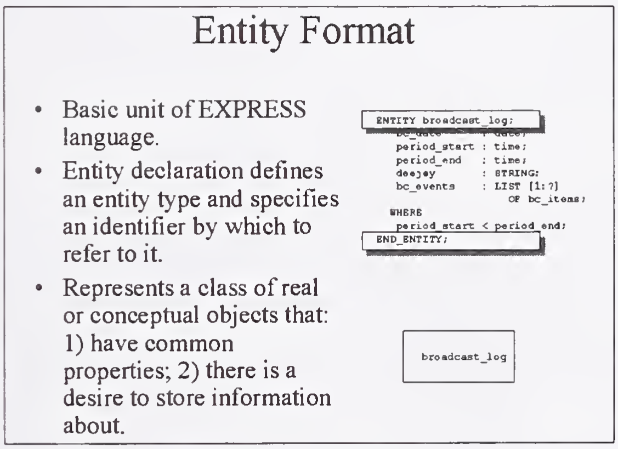


Figure 5-4: Entity

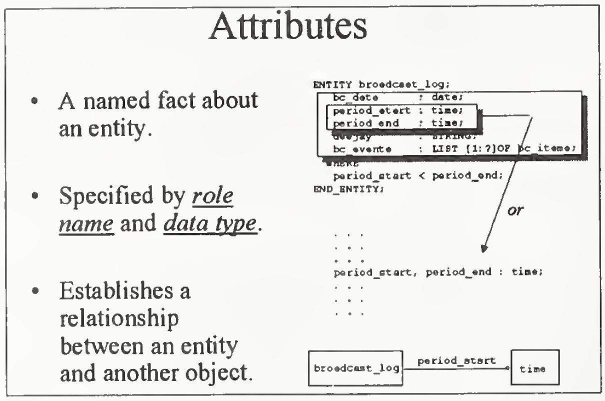


Figure 5-5: Attributes

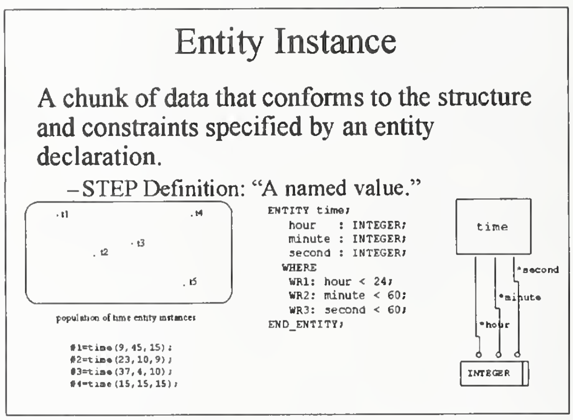


Figure 5-6: Entity Instance

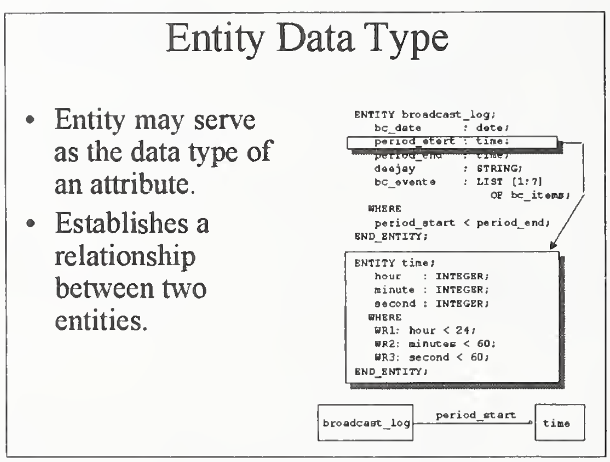


Figure 5-7: Entity Data Type

To demonstrate opportunity for ambiguity in EXPRESS, the following is an example of a sloppy definition. An entity of the data type `person` may have named attributes `last_name` and `age`, having values `Smith` and *32* respectively. This `Smith` entity is called an *entity instance*. It is an instance of the *entity data type* `person`.

A sloppy EXPRESS definition of the person entity data type might be:

```
ENTITY person;
  last_name : STRING;
  age : INTEGER;
END_ENTITY;
```

This is sloppy because, among other reasons, it disregards the fact that age must be a positive number, the choice of type INTEGER might not be wise and at any rate, INTEGER (if a unit of years was intended) should be limited to integers less than 200. Now one has a taste of what is meant by 'ambiguous'! Industrial-strength EXPRESS information models are far more rigorously defined.

Considering the above terminology Table 5.1 shows EXPRESS constraints on the data.

Constraint	Definition of Constraint	Example
attribute's type	the value of an instance's attribute must be type-compatible with the type declared in the entity's EXPRESS definition	
semantic consistency of entity instance	rules (informally called WHERE RULES) can be associated with the entity type	an entity of a type named unit_vector possessing REAL valued attributes x and y might have a where rule requiring x**2 + y**2 = 1
entity instance's type	EXPRESS provides language elements for defining and composing hierarchical entity types, that is, types that are related to each other by an 'is kind of' relationship	an abstract type person might have subtypes male and female. An instance of person must be one of male or female but not both
populations of entity instances	Populations of entities (datasets) must satisfy constraints described in global rules	Such rules can constrain the cardinality of the instances of a type (e.g. there is only one CEO)

Table 5-1: EXPRESS Constraints on Data

NIST was an early proponent and supporter of testing before standardization. The size and scope of the information models being developed for STEP needed to be contained in a rigorous manner. Testing was a means of achieving this rigor. The testing process involved tracing information models to real-world requirements in the form of data to support the model. This process often served to reduce the scope as no data was found to support the model. It also served to identify holes in the model where data was available but not connected to the information model [104]. In partnership with PDES, Inc., NIST provided software services to support early model validation. The National PDES Testbed used by PDES Inc. members in early testing activities was lovingly called the "PIG Pen" as many PDES, Inc. groups spent many hours there at NIST.

The initial software toolset served to validate the structure of the information models. It was thrown together rapidly out of existing ideas for implementing STEP and prototype software. It consisted of two main programs: QDES (the Quick and Dirty Editing System developed in Smalltalk) [105] and an EXPRESS-to-SQL translator [106]. QDES was used to create and massage data into a STEP format. The EXPRESS-to-SQL translator was used to create database tables for the data. SQL queries were written against those tables to validate whether the correct data could be reached using the developing information models. The software design and implementation had many drawbacks yet still served the purpose, even if very awkwardly [107][108].

In the early 1990s, NIST led an effort to create a more robust testing system that would better serve the testing needs [109]. The STEP Class Library (SCL) was developed as part of that effort under the heading of the Validation Testing System Project funded by the CALS Program [110]. With support from the Defense Advanced Research Projects Agency (DARPA), NIST continued developing the SCL for use by STEP developers and implementors.

The SCL is a set of C++ class libraries that are capable of representing information conforming to ISO 10303-11. The libraries may be used to build executable C++ applications, which make use of information contained in an EXPRESS file. They contain such features as a dictionary of EXPRESS schema information and functionality for representing and manipulating instances of EXPRESS objects. Simple applications, such as ones that read and write EXPRESS data in the form of ISO 10303-21[111] files, can be written easily and are included in the SCL release.

SCL was developed with several purposes in mind. Most notably, it has been useful for validating emerging concepts for STEP implementation methods and for developing software for STEP-based applications. Particular attention by NIST has been devoted to implementing the following ISO 10303 parts:

- ISO 10303-21:1994, Implementation methods: Clear text encoding of the exchange structure.
- ISO 10303-22: (to be published) Implementation method: Standard data access interface specification.
- ISO/DIS 10303-23, Implementation method: C++ language binding to the standard data access interface.
- ISO/CD 10303-26, Implementation method: Interface definition language binding to the standard data access interface [112].

Additional tools developed by NIST can identify EXPRESS violations in a dataset given the corresponding to those constraints found in Table 6.1 [113]. Such tools allow one to determine whether a STEP-capable product is emitting a 'valid' response. The question then is, how far do such constraints bring us toward ensuring unambiguous communication? Things can be consistent at one level and meaningless at another. One might intend, for example, to produce a one-centimeter cube with a STEP-capable CAD system and get instead a two-centimeter cube. This sort of error sometimes originates with an error in transposing the CAD system's internal data structures into STEP entities (for the purpose of the communication). This error is outside the purview of EXPRESS's validation role.

There are also situations in which the constraint does indeed fit into one of the categories above, yet it is simply too difficult to describe in EXPRESS. Examples are found in the modeling of curves and surfaces. It would be useful to indicate that closed curves exchanged between systems are recognized as being closed by both systems. Whether or not every closed curve entity is mathematically closed depends on the implementation of the curve objects on both systems.

Developing EXPRESS as a programming-like modeling language has actually proved a hindrance in developing the ARMs (though it is an appropriate language for AIMs). The reason is that the objective of ARM development is the discovery and "formalization" of the semantics or information found within a scoped domain. The programming and data-structure-like character of EXPRESS leads modelers to develop models that reveal their data processing experience, e.g., combining distinct concepts in a single entity with optional attributes. Models of the same domain developed in the textual form of EXPRESS will be very different from those developed in EXPRESS-G.

5.11 WHAT CAN BE GENERATED?

A formal modeling language is limited in what utility it provides for model validation. Likewise, there are limits to the utility of software generated from EXPRESS for exchanging data instances based on the model. The problem here is that the implementation generator can not anticipate what form the data structures should take. The application that would use this software presumably has its own data structures, different from those that the generated software might create. The data must be transposed from objects in the form defined by the generated software to the application's native form. Although this might not seem too difficult, it is in some sense the whole problem of unambiguous data exchange only restated as exchange between internal memory structures rather than file exchange. Finally, maintaining two sets of structures, those from the generated software and those from the application program, may present a demand on system resources that make the approach unattractive.

Despite the above concerns, generating software from EXPRESS provides a very significant improvement in software development productivity. Such software can seldom be used without modification, but it provides the developer with a good start.

5.12 MINOR ANNOYANCES

There has never been a computer language that has pleased all of the people all of the time. EXPRESS as a usable language has accomplished pleasing some of the people some of the time. EXPRESS has its quirks and nearly everyone who knows the language recognizes some of them. To mention a few...

- **WHERE rules on entity definitions may rely on populations.** WHERE rules on entity definitions are intended to define constraints on the state of the entity without reference to any population of entities. However, it is possible (and examples exist in STEP standards) for WHERE rules to call the EXPRESS built-in function USEDIN (that finds entities which reference the argument entity). When USEDIN is called from a WHERE rule, the intent of WHERE rules is violated.

- **Important information gets buried in procedural code.** EXPRESS is at a disadvantage relative to predicate calculus-based syntax. EXPRESS parses to relatively complex syntax trees that do not (because of their complexity) suggest a simple working form of the information. For example, a working form that enables deductive retrieval is far better suited for developing programs that could solve the difficult problems of information modeling. (An example of a difficult problem is what STEP developers call 'schema integration'.)

Even keeping its recognizably Pascal-like syntax, EXPRESS would benefit significantly from a few syntactic improvements. For example, automated manipulation of the language would be simpler had there been language elements to represent type and schema constants. EXPRESS uses strings in these situations. Thus programs have to guess whether or not an arbitrary string refers to a schema or type.

Finally in this category, EXPRESS would benefit from a built-in function with the meaning of the predicate calculus quantifiers (e.g., the existential quantifier, "there exists"). In practice, EXPRESS rules are replete with the notions of "there exists" and "there does not exist" coded in a form such as SIZEOF (QUERY | NOT....) = 0, which is a very awkward way to say, "there does not exist."

The operator NOT should be permitted in the supertype clause. The supertype clause defines the possible combinations of complex types that include the subject entity. In the current syntax it is impossible to state that a combination is not permitted, this task must be relegated to procedural code in the entity's WHERE rules. In this sense, this issue is another example of burying important information in procedural code.

5.13 CONCLUSION

Modeling is an essential part of the STEP architecture and methodology. Due to the very nature of product data requirements, the standard's process, and the difficulty of the tasks, STEP modeling exhibits great innovations and serious challenges simultaneously. Of course, modeling is certainly possible today and the STEP community continues to develop models furiously. STEP modeling remains a difficult field requiring experience that few practitioners have, and still fewer have mastered.

EXPRESS provides the standard parts of ISO 10303 with a powerful tool by which information models can be described and corresponding datasets validated. The language has a Pascal-like syntax that is familiar to many programmers. Although not perfect, EXPRESS has easily proven its worth to the STEP development community. Other developing standards within ISO TC 184/SC4 are using EXPRESS (ISO 13584 and ISO 15926). Other ISO technical committees are also considering the merits of EXPRESS, for example, as part of the development efforts in ISO TC 211, the Technical Committee on Geographic information and geomatics.

CHAPTER 6

SHARING VERSUS EXCHANGING DATA

6.1 INTRODUCTION

The concept of exchanging product data among the software systems used in manufacturing enterprises is central to STEP. While product data exchange has been a fundamental goal for the standard since the development effort initiated, there has also been a desire to enable product data sharing among software systems. The question is what is meant by the concept of product data sharing? To answer that, the concept of data exchange must first be defined in more detail so that the distinguishing characteristics of the two concepts can be identified. Implementation levels and exchange versus sharing were introduced in Chapter 3; the concepts are elaborated upon here.

6.2 DATA EXCHANGE IN THE CONTEXT OF STEP

Data exchange is the transfer of information from one software system to another via a medium that represents the state of the information at a single point in time. This information snapshot is encoded digitally, typically in an ASCII or binary representation. Figure 6-1 provides an example of data exchange. When one receives a monthly bank account statement, the information from the bank to the customer represents data presented at a single point in time.

The information is provided in an electronic file. This allows management by computer operating systems. Transmittal from one computer to another is via portable storage media, distributed file systems, electronic mail, and by numerous other network communication mechanisms. Representing and transmitting information are necessary but not sufficient for meaningful data exchange among software systems. Interpreting the information is the more challenging aspect of data exchange. Previous chapters have described various STEP technologies that, when taken together, are intended to enable common interpretation of information among software systems. For the purposes of this discussion, the focus is on how that interpretation is realized in practice.

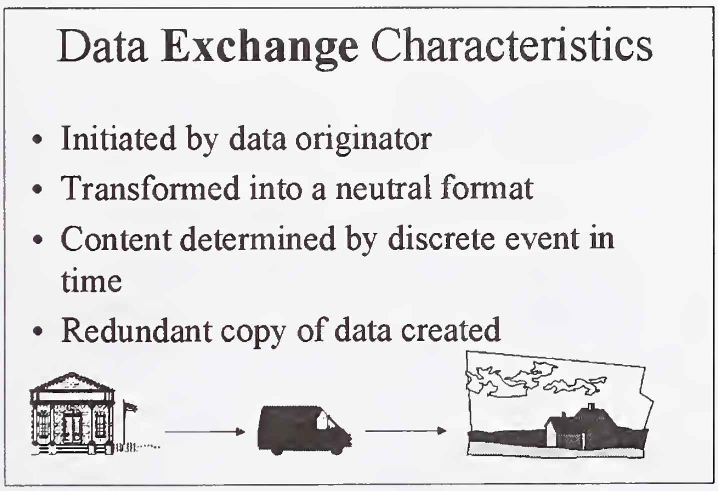


Figure 6-1: Data Exchange

For a given software system (here referred to as System A) to generate a data exchange file, System A must implement specific functionality generating a file that represents the information to be exchanged. In practice, this functionality is typically referred to as a capability for file export. This file-export capability is a translator that maps the internal representation of the information to be exchanged to an external file. If the exchange file is System A's proprietary, or native format, the translation is likely a matter of structuring the file encoding of the information for efficiency and not of generating different representations of the information. If the structure and content of the exchange file is not defined by System A but rather by some other party (i.e., a "neutral" exchange file), then generating the exchange file involves transforming System A's internal representations into those specified for the exchange file. How accurately these transformations reflect the information as it was originally represented internally in System A depend on:

- How compatible or equivalent the representations needed by the neutral exchange file are with those of System A.
- How well the translation software was implemented.

Consider another software system (System B) that receives the neutral exchange file created by System A. For System B, the process described above happens in reverse. System B will have to provide functionality in its implementation for accessing the neutral file as well as for interpreting the contents and creating an internal representation of that information. The first stage, accessing the file, is typically referred to as importing the file. The second stage, interpretation, is a translation process whose accuracy again depends on the compatibility or equivalence of the neutral file and System B's representations along with the quality of the translator's implementation. Assuming the translation is completed accurately, System B will have an internal representation of the information provided by System A at the time of its export through the neutral exchange file mechanism. Whether System B's internal representation of the information provided by System A is equivalent to System A's internal representation is a topic covered as interoperability testing in Chapter 8. At best, we can say that System B's internal representation is equivalent to the information provided by System A.

6.3 DATA SHARING IN THE CONTEXT OF STEP

Data sharing provides a single logical information source to which multiple software systems have access. Controls over access to the information, updates to the information, ownership of the information, and so on, are typically provided in implementing and administering the information source. Using the banking example in Figure 6-1, Figure 6-2 shows data sharing of the same information.

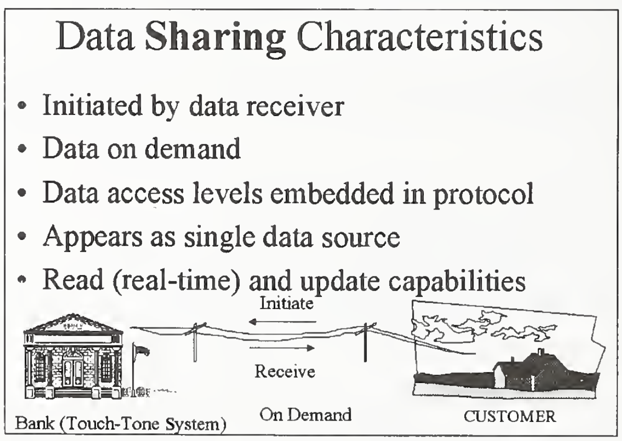


Figure 6-2: Data Sharing Characteristics

The information source may be realized as a database management system, a specialized file system, or a combination of the two. Product Data Management (PDM) software systems typify the state-of-the-practice with respect to the functionality provided for information ownership, revision maintenance, and information access. By their nature, PDM systems manage data sharing at the product level, that is, the smallest unit of access is a product representation. The mechanism used for a product representation is typically some type of digitally encoded file (it could be a neutral data exchange file or a proprietary format file). Software systems interfacing with a PDM-style system still face the issues of translation and interpretation of the exchange files. Data sharing detailed at a finer level than that of a product requires that the information source support the creation, access, and manipulation of the underlying data elements. These data elements constitute the product data representations -- not to mention handling the issues of access control during data revisions. Data sharing at this finer level of detail is desirable because it enables the close coupling of applications that create and manipulate data with the single logical resource maintaining access to the information. Ideally, the information resource maintains the primary copy of the information, thereby eliminating the need for the applications to maintain their own, local copies of the data while it is in use. While in theory this may be the ideal situation, in the context of STEP this would require major re-engineering of many of the software systems that STEP intends to support. This re-engineering is due to the differences between the representations STEP requires and those existing in the software systems.

6.4 SHARING DATA & EXCHANGING DATA COMPARED

The characteristics that distinguish data sharing from data exchange are centrality of the data and ownership of that data. In the exchange model, the software system maintains the master copy of the data internally and exports a snapshot of the data for others to use. This use is without explicit controls on how changes to the internal data are made current with the exchange file version and vice versa. Any other software system that imports the exchange file has effectively assumed ownership of the data. Such data is not synchronized with respect to the "master" copy of the data contained in the originating software system. Hence software systems are not sharing data because the only safe assumption is that each software system has a different version of the data.

In the sharing model, there is centralized control of ownership and there is a known master copy of the data (the copy maintained by the information resource). The master copy of the data can be accessed and revised under controlled circumstances. Currency of the data can be enforced to ensure that all software systems have access to the same data. The revision history of the master copy of the data can be tracked; the currency of a software system's copy of the data can be verified.

In theory, the data-sharing model alleviates the revision control problems associated with the data exchange model. As such, data sharing is an ideal for which to strive. The STEP community recognized early the need to distinguish such capabilities by implementation levels.

6.5 STEP IMPLEMENTATION LEVELS

In 1988, an ad-hoc group within the IPO led by Ontologic, attempted to define the nature of STEP implementations envisioned. This ad-hoc group, introduced in Chapter 3, informally identified four kinds of expected implementations based on the technology at the time. Of the different levels defined in Chapter 3, Level 1 was the only implementation mechanism attempted for the initial release of ISO 10303. It is also the principal implementation mechanism in use today.

The STEP committees separated the specification of implementation mechanisms from the specification of the data. By virtue of having the formal, computer-interpretable EXPRESS language [114] as the means for specifying the data, STEP implementation mechanisms are made independent of any particular data specifications. Hence, the challenge for the group participating indeveloping what became known as ISO 10303-21, Clear Text Encoding of the Exchange File [115], was to devise a specification for an exchange file that was derived solely from EXPRESS.

Work on developing the exchange file specification began in the 1986 timeframe and included significant participation from Boeing, the German Nuclear Research Center (KfK), McDonnell-Douglas, NIST, Rutherford Appleton Laboratory, Structural Dynamics Research Corporation (SDRC), and numerous others. There was a clear intention to provide a file structure that was better than that realized in IGES. IGES files derived their structure from the era of Hollerith cards: field and record-oriented. IGES files were not readily interpretable to humans, and given the amount of manual analysis needed to identify exchange problems, there was a certain impetus to make the STEP exchange file structure more easily interpretable to humans. Consequently, it was satisfying to realize the file structure as a text-style file. However, there was also the potentially conflicting objective to devise a file structure that was more compact than IGES. The compactness objective argued against the use of a text file structure and argued for alternatives such as a binary encoding. In the end, the text encoding won out, with the caveat to develop alternative encodings as interest warranted (as had been done with the Computer Graphics Metafile standard [116]). There have not been any alternative encodings pursued as of this writing, because file compression software tools have mitigated the issue of compactness.[24]

Undoubtedly the biggest challenge in the process was handling the capabilities EXPRESS provided for inheritance among entities. For the "single inheritance" case, wherein an entity inherits attributes from a single ancestor, there was little difficulty in devising a mapping for the exchange file. The "complex inheritance" case proved troublesome. With multiple inheritance, an entity instantiation is essentially that of multiple types simultaneously inheriting attributes from multiple ancestors. The mapping to the exchange file required developing an algorithm to convey how the exchange file manifestations of such instances would provide sufficient information in the file. This would allow the interpreting software to determine to what ancestral entities the attributes belonged, and therefore, with how the attributes should be dealt. Many of the earlier EXRESS models developed in STEP did not make use of the "complex inheritance" feature; later models have.

As the exchange file mapping was being developed concurrently with the EXPRESS language, there were many situations where the exchange file developers found themselves out of sync with the language developers. At one point during the evolution of EXPRESS, the so-called "default" model of inheritance was reversed completely. This rendered all existing EXPRESS models incorrect, along with the exchange file mapping; hence, all prototype exchange files and the experimental software tools that had been developed by NIST to interpret EXPRESS and exchange files were rendered immediately obsolete! Such were the problems when dealing with a language that was developed in parallel.

Over the years, much work has focused on the implementation levels as described in Chapter 3. Particularly, many issues have been associated with trying to implement levels two and three, [117] and to a lesser extent level four. STEP's Standard Data Access Interface (SDAI) [118] is the result of these efforts. SDAI covers some aspects of the original visions [119] for both levels 2 and 3. The need to allow freedom and flexibility to implementations while leveraging existing technology drove the initial scoping of SDAI. To date, there are no known level four implementations.

6.6 SDAI EVOLUTION

PDES, Inc. and its member organizations at that time, notably -- NIST, IBM, Digital Equipment Corporation (DEC), Electric Boat Corporation,[25] STEP Tools Inc., and SDRC -- spearheaded the ISO project to establish a programmatic interface for STEP data. PDES, Inc. members participated in the ISO working group developing SDAI and conducted significant prototyping activities in the area. Abroad, Europe's ESPRIT IMPPACT project contributed to developing SDAI and in particular specified significant portions of the data dictionary. Many other countries (notably Germany, Japan, and the United Kingdom) also hosted contributing research projects.

Early ideas for SDAI envisioned an interface to a relational database management system with a standard interface such as SQL [120]. Experimentation with such an interface found that the mapping of engineering data into the relational systems did not provide acceptable performance. It was also not easy for application protocol developers

---

[24] An algorithm for generating shortened names from the normative STEP entities was developed by Rutherford Appleton Laboratory. A program that implements the algorithm is executed at NIST as part of administering the continuing standards development process. The abbreviations resulting from the program are included in the standard documents and are maintained in a registry at NIST. The net effect of the process is to reduce the size of the STEP data exchange files that use the abbreviated entity names. In ISO 10303 the short names must be used. Other SC4 standards are also requiring short names, e.g., ISO 13549 Parts Library.

[25] Formerly General Dynamics Electric Boat

to develop tests of their information models based on these systems. These results led to an investigation of object-oriented database systems as the repository to provide an SDAI.

The original SDAI specification drafted by NIST was presented to industry along with CAM-I's Application Interface Specification (AIS), at a joint workshop in St. Louis in 1990. The workshop highlighted the need for both early and late bound interfaces and resulted in changes to the SDAI specification, which were reflected in the next version of the draft. This document was the focus of a significant prototyping effort undertaken by PDES Inc. and led by NIST. The prototyping activity involved several object-oriented database vendors (e.g., Versant and Objectivity) as well as DEC and SDRC. The strength of the prototyping results (and the vendor commitments to the prototyping efforts) was instrumental in attracting significant attention to the SDAI specification. With this attention came additional interest in the SDAI Working Group's activities. To keep with the SC4 goal of ISO 10303 being independent of implementation methods, the specification of bindings to particular programming languages spun off as separate projects from the functional specification of SDAI. While this helped to keep the functional specification focused, progress on the standardization remained slow. This was due to the increased number of participants in the Working Group and the numerous issues surrounding the use of a general-purpose interface for application-specific standards. The Working Group was mired in arguments over compatibility of SDAI with the exchange file format, support for interoperability between APs, and the execution model of SDAI.

The four levels of STEP implementations presented in 1988 did not predict the capabilities for distributed computing available today. More recent work on STEP implementations has focused on solutions that leverage these capabilities. More specifically, members of the National Industrial Information Infrastructure Protocol (NIIIP) consortium (NIST, General Dynamics, STEP Tools Inc., IBM) have pushed the SDAI Interface Definition Language (IDL)[121][122] binding specification with early prototypes demonstrating its usefulness for virtual enterprises. Additionally, NIIIP provided significant input into a new ISO work item to specify the mapping language known as EXPRESS-X (described in Chapters 5 and 10) and a binding of SDAI to Java [123]. Both of these emerging standards are geared to support the distribution of data.

6.7 SDAI --INTENDED PURPOSE

SDAI provides an Application Programming Interface (API) to data described by an EXPRESS information model. The specification of a standard API supports the decoupling of the data exchange software from the application that will import or export the data thus optimizing the usefulness of the data exchange software. The API bridges between an application and the software used to store and manage the data (Figure 6-3[124]).

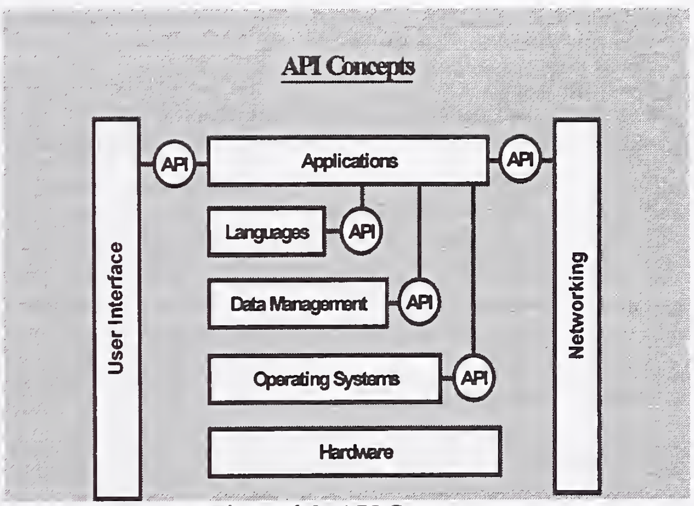


Figure 6-3: API Concepts

The data on disk may be represented in a number of ways, one of which is ASCII files formatted as described in 10303-21. The data may be stored alternatively using more advanced database software. The designers of SDAI provided enough flexibility in the interface to support storage within a database, while still allowing for the possibility of file-based storage.

In many ways SDAI resembles the interfaces of traditional database management systems such as SQL [125] or CODYSAL [126]. What distinguishes SDAI from these other database interfaces is that, taken in context with the rest of ISO 10303, it defines a semantics-based interface. In contrast, traditional database standards only define a mechanism for access to anonymous data. Put another way, SDAI defines a standard view of data based on an EXPRESS model and not on how the data are actually represented.

Combined with the rest of ISO 10303, SDAI is similar to domain-specific interfaces such as CAM-I's AIS; it provides a domain-specific interface to data. However, the AIS interface provides more than a view of data, it specifies functions on the data. While this capability has long been requested for STEP, it has yet to be defined.

Another significant difference between SDAI and SQL is in the style of data access. SQL was designed for record-oriented data with a limited number of data types. The data described by STEP forms a network of interconnected, complex data types with relatively few instances of each one [127][128]. Traversal of the links, rather than bulk processing, is the more predominant use for this data. The relational database paradigm does not support complex traversals in a manner that is intuitive to use. STEP's data access needs are more compatible with the representational capabilities of object-oriented database systems [129].

6.8 SDAI AND ITS FAMILY OF STANDARDS

SDAI really requires a combination of standards. The first in the series, ISO 10303-22 [130], is a functional specification; it is independent of a programming language. ISO 10303-23 [131], -24 [132], and –25 [133] bind SDAI to particular programming languages: C++, C, and FORTRAN respectively. Work to develop the Fortran binding was eventually abandoned for lack of interest.

Since work on 10303-23, -24, -25 began, efforts have emerged from the software community to define a more general purpose programming language binding to be used by APIs. In particular, the software community is standardizing an IDL binding [134]. A motivating factor for ISO 10303-26, the binding of SDAI to IDL, was to eliminate the need for further language bindings. In practice, this did not happen, and recently, work has begun on a Java binding.

The first language binding to be promoted as an international standard was the C++ binding, ISO 10303-23. Much effort went into making the C++ binding compatible with existing object-oriented database systems. NIST and others throughout the world (notably in Germany, Japan, United States, and United Kingdom) developed prototypes of such systems.

Beyond language bindings, another distinguishing feature of the SDAI interface is that of early versus late binding style. An early bound interface maps the EXPRESS language directly into programming language constructs thereby allowing the data model to be bound to the interface at compile time. A late bound interface is dictionary-driven in that the application uses a run-time dictionary to access the information model. Late bound applications can change data models at run-time (Figure 6-4). The C++ and IDL bindings allow for, but do not require, both types of interfaces; the C binding specifies only a late bound interface.

6.9 SDAI'S COMPATIBILITY WITH EXPRESS

SDAI is intended for accessing and manipulating data instances created according to an EXPRESS schema. Conceptually, SDAI consists of two parts: the SDAI schemas and the mapping of EXPRESS to the target language constructs.[26] The SDAI schemas specify constructs needed to run SDAI independent of the data that is being accessed.

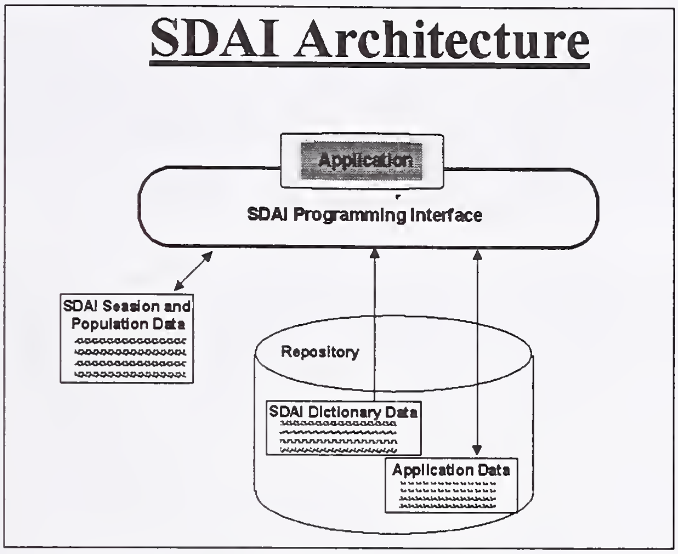


Figure 6-4: SDAI Architecture

An EXPRESS information model represents SDAI in two different ways. First, it may be represented through the data dictionary; secondly, it may be represented directly in data structures available in the programming language. ISO 10303-22 describes the contents of the data dictionary. The dictionary does not capture all of the EXPRESS language but rather captures the core data structures of EXPRESS along with a limited number of explicit constraints.

The thoroughness of a mapping of EXPRESS to a particular programming language will be different depending on the binding language. Many of the EXPRESS constructs are not represented directly in a given programming language making compile-time checking of an application impossible. For instance, the EXPRESS type system does not map easily to C; therefore, the C language binding does not include an early bound mapping. On the other hand, the C++ type system is rich enough to cover a subset of the EXPRESS type system and is applied where possible. Where the type systems differ significantly C++ compiler's type-checking capabilities can not be used and run-time type-checking must be employed.

SDAI's language bindings define the specifics of an API. The different language bindings vary in their support of the semantics of EXPRESS based on the representational capabilities of the language. The C++ APIs resulting from ISO 10303-23 and 10303-26 highlight such differences. ISO 10303-23 is able to leverage the capabilities of the C++ language to provide robust support for many EXPRESS constructs [135]. The C++ API resulting from ISO 10303-26 is minimal in its support for many of the EXPRESS constructs. For example, EXPRESS's array type may be represented in C++ as a class that supports a minimal interface (such as that required for C-style arrays), as well as a richer interface that is able to safely handle array bounds and other constraints on the type. On the other hand, the SDAI IDL binding specifies a more minimal interface for the representation of arrays in the form of an IDL sequence type. An IDL sequence would not pass information along to the application, such as the bounds of the array, nor does it allow the application to test whether elements of the array have been set or not.

Where a programming language does not provide direct support for EXPRESS constructs, enforcing some of these constructs is, nevertheless, the responsibility of the SDAI implementation. Type checking is one such example.

[26] Only those specifications that provide for an early binding describe the mapping of EXPRESS to the binding language.

Also, compile time support for bounds on arrays is not supported by any of the languages bound to SDAI; however, a function is defined in SDAI that allows an application to trigger a check for whether an array honors the bounds set for it.

The EXPRESS FUNCTION construct is not supported in the SDAI language bindings. SDAI provides hooks for an application to trigger functions from the information model. The binding of functions to a programming language is outside the scope of SDAI and was not undertaken in any of the current binding documents.

6.10 SDAI SUPPORT FOR APPLICATION PROTOCOLS

The inter-schema referencing capability is perhaps the most prominent feature of EXPRESS not provided in SDAI. A premise given for an SDAI implementation is that the schema supported by an instance of the interface is based on the "completely expanded form of a schema" [136], or the so called "long-form." The schema for an instance of SDAI is a schema that fully expands all the references by assuming the EXPRESS definitions are contained within the schema. Information about the originating schema is lost with respect to data instances and, presumably also, to the data dictionary; however, the SDAI dictionary provides minimal support for multiple schemas. The dictionary accommodates multiple APs where each AP is represented as a single schema (a long-form schema). It does not address the fact that two AP schemas may be derived from the same underlying resource parts.

6.11 CONTRASTING ISO 10303-21 WITH ISO 10303-22

ISO 10303-22 describes an exchange format just as ISO 10303-21[137] does, but most of the similarity between the two standards ends there. 10303-21 describes an ASCII representation, it does not provide any information about state, and it only represents a snapshot in time. SDAI provides data in a format usable by the application without format conversion. When used with a shared data repository, SDAI supports application access to dynamically changing data. Additionally, SDAI supports changing data such that the data moves in and out of complete and consistent states. SDAI specifies functions that allow an application to initiate when to check the data for completeness and consistency.

One requirement imposed on the developing ISO 10303-22 was that an SDAI application should be able to produce a 10303-21 exchange file from the data repository. As such 10303-22 is upwardly compatible with 10303-21.

A particularly noteworthy construct contained in both parts is the SCOPE construct. The SCOPE concept defines a "scope of reference and existence relationships" [138]. This construct only appears in the standard's implementation specifications. It does not appear in EXPRESS or in the information models themselves since it is a concept that applies specifically to instances of data. Perhaps it is more interesting to note that in practice, many U.S. users of the standards ignore this feature and it is used primarily within Europe. The value of retaining this concept is still under debate.

6.12 IMPLEMENTATION CLASSES IN ISO 10303-22

ISO 10303-22 defines six implementation classes based on five characteristics (Table 6-1). The lowest implementation class provides minimal support for data exchange and is essentially an API to an exchange file. The higher classes provide progressively more sophisticated features, culminating in rich support for evaluating EXPRESS expressions. Such rich support enables complete constraint checking and calculation of derived attributes.

Implementation Class	Transactions	Expressions	Session Record	Scope	Domain Equivalence
1	none	none	no	no	no
2	none	simple	no	yes	no
3	none	complex	no	yes	yes
4	model	simple	yes	no	no
5	full	simple	yes	no	yes
6	full	complex	yes	yes	yes

Table 6-1: ISO 10303-22 Implementation Classes

While supporting system evolution, SDAI's implementation classes also provide guidance for software modularization. As applications adapt to newer SDAI implementations, more of the functionality required by the application will be resident in the SDAI implementation. Bearing this in mind, applications based on SDAI should be designed such that they will not be impacted adversely by the evolution.

6.13 TESTING FOR SDAI

Work has been initiated on ISO 10303-35 [139] to define conformance test methods for SDAI implementations. Much development still needs to be done in this area. Little activity has focussed on the need to test conformance to a shared database containing data about multiple products. SDAI's support for long-transactions further complicates matters with respect to testing. Without testing, however, the usability of SDAI systems will be restricted. Much more will be said about testing in Chapter 8.

Each part of STEP, including SDAI, defines several conformance levels. Additionally, a system using SDAI builds on not only 10303-22 and one of the language bindings specified in 10303-23, -24, or -26, but also one or more application protocols. All of this must be laid over an equally, if not more complex, computing infrastructure made up of a wide variety of hardware platforms, networks and firewalls, and application software. Flexible assembly and evolution of open systems using STEP will require strong support for conformance and interoperability testing. The definition methods by which this will be accomplished are still an open area for research.

6.14 CONCLUSION

Although SDAI is only a component necessary for data sharing, it is today's best answer to product data sharing within a STEP environment. Some of SDAI's current limitations include nonsupport of:

- Concurrent access to multiple SDAI sessions by multiple applications.
- Connections to remote repositories.
- Access or manipulation based on semantics of data.
- Specification of data formats.
- Creation, deletion, or naming of repositories.

However, even with its current limitations, SDAI:

- Specifies a standard access mechanism to databases (repositories) that can be specified in EXPRESS.
- Separates the interface from the binding to a particular programming language.
- Permits application system independence from data storage technologies.
- Has an object-oriented view of data, regardless of implementation.

The goal of product data sharing, while offering many benefits over data exchange, can only be accomplished in a restricted setting today. Although minimal attention has yet been paid to the application of knowledgebase technology, the exploding use of the Internet and related technologies have redirected STEP implementation efforts to address issues surrounding distributed access. Chapter 10 shares some exciting thoughts on where SDAI and supporting technologies may lead the STEP community in the future.

CHAPTER 7

THE USER PERSPECTIVE

7.1 BACKGROUND ON APPLICATION PROTOCOLS

Chapter 2 mentioned that the initial utterance about Application Protocols (APs) started with a U.S. Navy-sponsored NIST project. As part of the project for the Naval Facilities Engineering Command, NIST documented the limitations of the use of IGES for the Architecture, Engineering, and Construction (AEC) industries. This effort recommended to the Navy the development of IGES application protocols to support the high priority information exchange requirements of the AEC industries [140]. The current IGES subsets were insufficient for effective product data exchange. A fundamental premise of this recommendation was that information exchange standards must include the definition of the semantics of the information to be exchanged and the mapping of these semantics to the data structures for representing the information. NIST submitted these results to ISO TC184/SC4 for inclusion in the STEP project. This landmark approach in 1988 was a philosophical change to the way product data standards were developed to support industry.

Events prior to 1988 had indicated that additional constraints were needed to achieve desired, near-perfect levels of data transfer fidelity. The U.S. Department of Energy (DOE) had developed a "filtering" program as part of a DOE project, aimed at unambiguous transfer of mission-critical designs. Although the product's domain was reasonably well constrained for this project, software was developed that would detect entity use that could lead to inaccurate transfer between CAD systems. Members of the IPO who had participated on this project identified the notions about product domains as central to the emerging concept of application protocols.

As is apparent in the history recounted in Chapter 3, the recommendations to create APs were embraced, but the components and methodologies to develop APs were longer in the formulation. Once established as part of the STEP architecture, the next step was to provide some semblance of order to the many existing AP initiatives. In May 1990, SC4 requested member countries to select their top three priorities for STEP AP projects from eighteen existing proposals. The recommended criteria for this selection were:

- Is it feasible to complete the AP within one year.
- Does the AP meet an existing international industrial need.
- Is it feasible the AP needed can provide the resources for the first edition of ISO 10303.
- Is it feasible to implement the AP.

SC4 used the results of this survey to establish the first five AP projects. Since then, industry programs continued to propose AP projects, which are then reviewed and approved by SC4 as ISO 10303 AP projects.

7.2 PURPOSE AND PRINCIPLES OF APPLICATION PROTOCOLS

As mentioned in Chapter 4, the application protocol methodology is fundamental to the architecture and use of STEP. The AP methodology provides:

- The means to define industry requirements and to ensure that these requirements are fulfilled by STEP standards.
- The means of extending the STEP integrated resources to address new application requirements in a consistent manner.
- The means to validate the application protocol and to ensure that implementations are testable.
- The STEP vendor with a specification that can be used in developing useful and reliable software products.
- A useful scoping mechanism for a particular industrial domain.

- An effective means to document an industry's semantics.
- The basis for conformance testing of STEP implementations.

An application protocol is the part of ISO 10303 that defines the context and scope for the use of product data and specifies the interpretation of the STEP integrated resources in that context to satisfy an industrial need. Additionally, an AP enumerates the conformance requirements and conformance classes for implementations of the AP. The design of application protocols permits the reuse of STEP integrated resource constructs to ensure consistent implementations and the exchange of relevant data among diverse computer applications.

7.3 COMPONENTS OF AN APPLICATION PROTOCOL

The five major components of a STEP AP were introduced in Chapter 4, and are the [141]:

- Application context, scope, and application activity model.
- Information requirements and the corresponding application reference model.
- Mapping table.
- Application interpreted model that specifies the use of the integrated resource constructs to represent the application information.
- Conformance requirements for implementations of the AP.

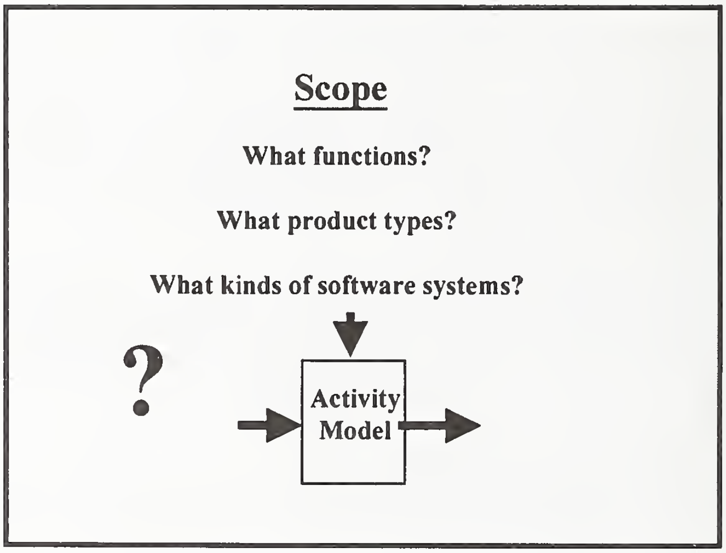


Figure 7-1: Application Protocol Scope

The scope of an AP describes the functionality and information that are accommodated by the AP. The scope of an AP is defined by the:

- Type of product.
- Supported stages in the lifecycle of the product.
- Required types of product data.
- Supported uses of the product data.
- Disciplines that use the product data.

The application activity model (AAM) is developed to establish understanding of the application tasks, processes, and the information flow of the application domain. The AAM identifies which information and information flows are within the scope of the AP, serving as requirements' gathering and a scoping tool. The AAM is described using the modeling capabilities offered by IDEF0.

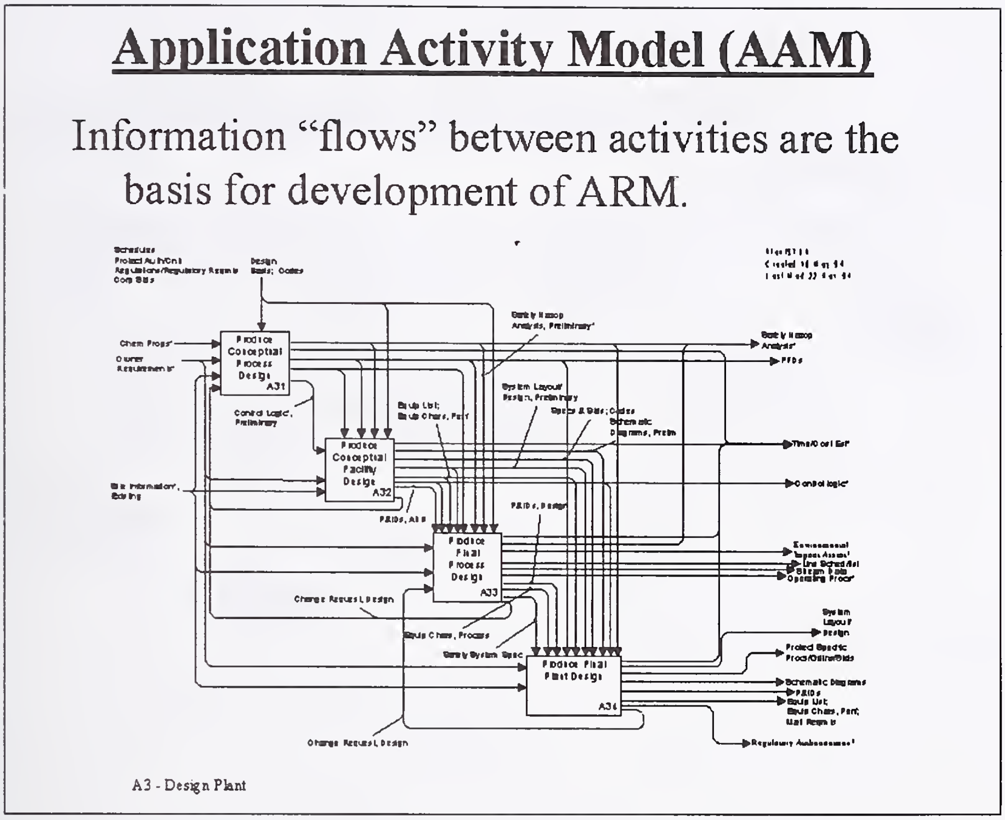


Figure 7-2: Application Activity Model

The AAM in Figure 7-2[27] shows the processes of interest to the AP. The information flows among activities are used as a scoping mechanism. Those outside the scope of interest of the AP are marked with an asterisk. The in-scope information flows serve as the source for information requirements. The AAM shows information flows at a level of detail that is useful for requirements gathering. The label on each flow indicates a real-world document or parcel of information; the information implied by the flow is broken down or enumerated. This breakdown then serves as a basis for developing the ARM.

The application reference model (ARM) captures the information that is most important within the application context. The AP developers determine what the AP must be able to "say." The information gathering for the ARM usually involves workshops where the functional experts participate and define the entities in terms that they understand. The modeling language chosen to represent the ARM is less important than the semantics of the model; the definitions of the objects in the model are the most critical aspects of the AP. The ARM formally describes the information requirements, structure, and constraints of the application domain.

Figure 7-3 correlates a fragment of an ARM to industrial requirements, in this case, a building section. The illustration shows the relationship between an object in the ARM and information conveyed by the drawing. Here, the important information is the fact that the wall is part of a building section (shown), that it has dimensions, and that it is, in fact, a wall.

[27] 10303-225 content was used to define these sample figures for the components of an AP.

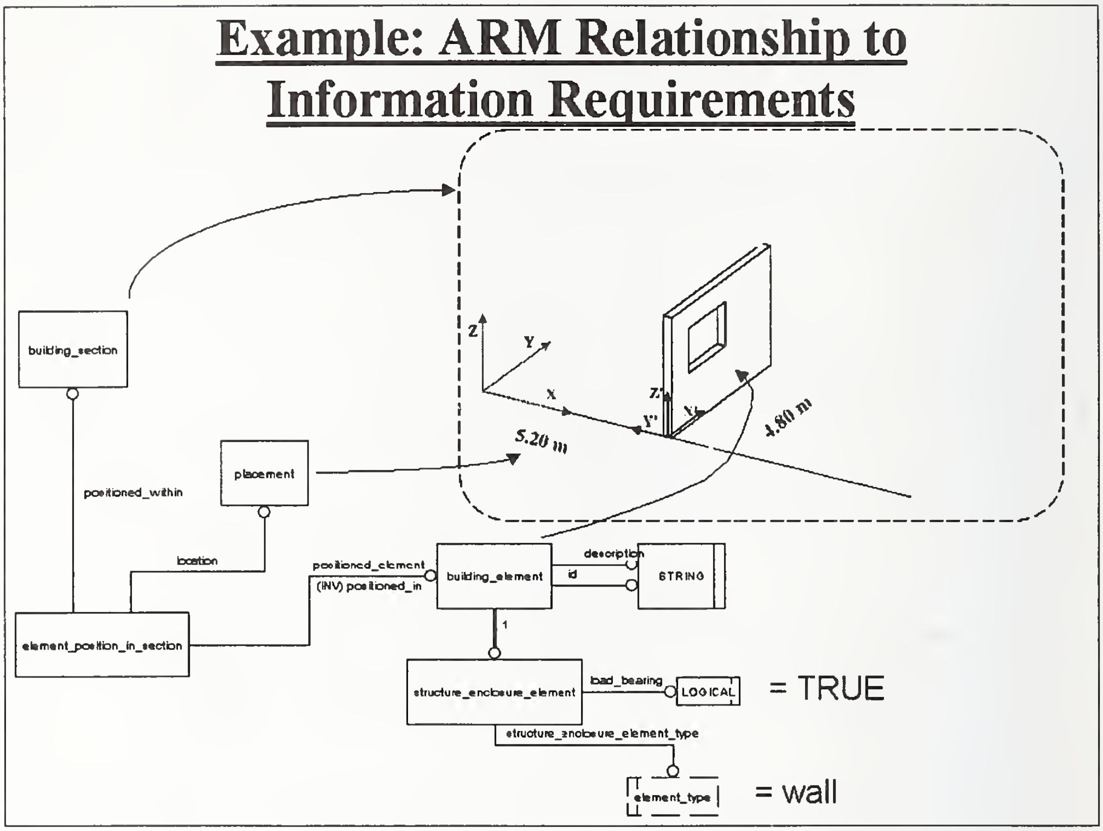


Figure 7-3: Application Protocol ARM

The application interpreted model (AIM) has several functions. It specifies the subset of the IR vocabulary used with the AP. It specifies the semantics of the generic data structure and adds constraints, thus providing the exchange specification for the AP. The requirements documented in an ARM are met through a selection of generic data constructs from the integrated resources and the specialization and constraint of these resources to meet the information needs of the ARM. Bringing IR entities into the AIM preserves the relationships and structure that the entities had in the IRs. Within the AIM itself, the entities may not be modified in any way that *changes the data structure in comparison to the IRs.* Each entity or its subtype still has the same number of attributes, and the attributes still appear in the same order. Constraints only affect the permissible values of an attribute. The fact that all APs use the same set of integrated resources means that all APs share the same fundamental underlying structure and semantics. All APs are related through the structure of the integrated resources. The AIM is constructed with the integrated resource constructs using EXPRESS mechanisms defined in ISO 10303-11. These mechanisms allow for the direct use of integrated resource constructs and their refinement.

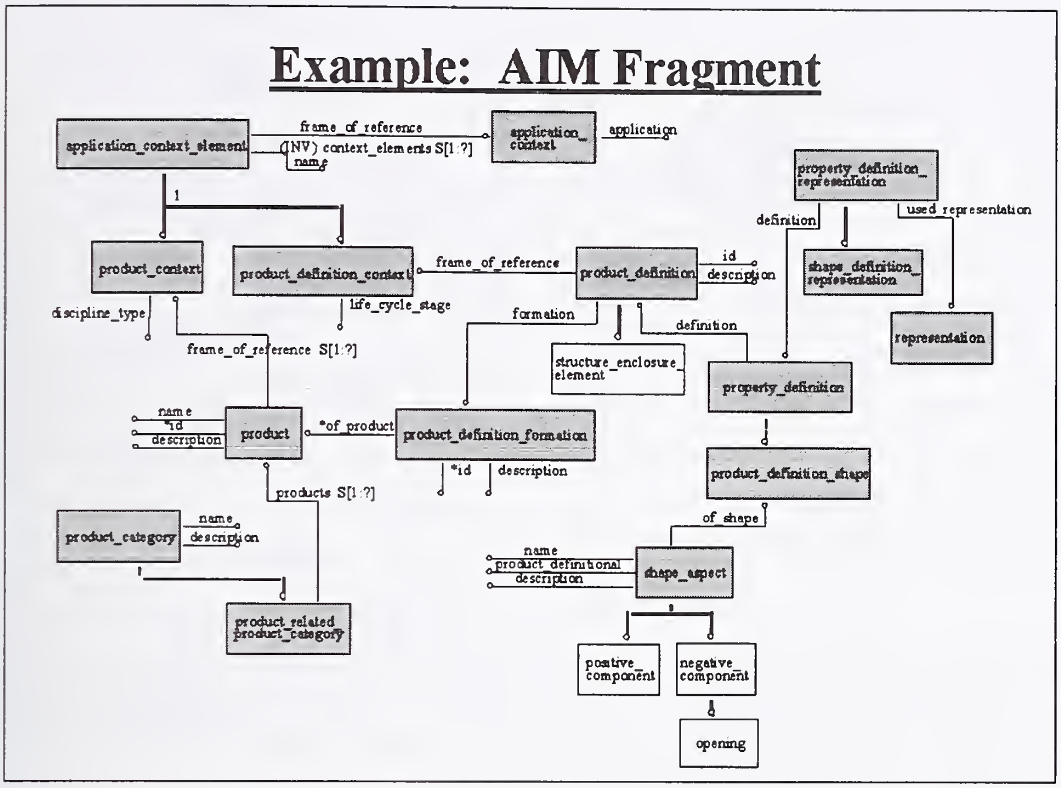


Figure 7-4: Application Protocol AIM

This EXPRESS-G diagram in Figure 7-4 is a fragment of an AIM. It shows the AIM subtypes. There is a subtype of product_definition called structure_enclosure_element. This subtype would be instantiated in a physical file to represent a wall (which is, after all, a structure or enclosure building element.) There are two subtypes of shape_aspect – the positive_component and the negative_component; negative_component also has a subtype called opening. An instance corresponding to opening would be created for the window in the wall; a window is an opening, which is a negative_component kind of shape_aspect of the shape property of a product_definition, which in this case happens to be a wall. This diagram may be complicated and difficult to understand for someone not involved in STEP. Its use here is intended to provide some idea of how an AIM works and how it relates back to the user information requirements.

The formal relationship between the ARM and the AIM is specified in the Mapping Table. The Mapping Table is the part of the AP that really ties the requirements and the exchange specification together. It contains rules and constraints specifying the use of the integrated resources for the AP. The combination of the AIM and mapping table completely specify the use of the generic structures for the AP. The mapping table describes how each information requirement is represented with STEP integrated resource constructs. The AIM is developed from the mapping table and specifies the schema, which uses the STEP IRs to satisfy the requirements documented in the ARM (Figure 7-5).

Example: Mapping Table (simplified)

ARM	AIM
structure_enclosure_ element	structure_enclosure_element <- product_definition
structure_enclosure_ element_type	product_category.name *where the .name attribute is "wall"*
load_bearing	property_definition *where the .description attribute is "load_bearing"*
component_ location	mapped_item

Figure 7-5: Application Protocol Mapping Table

In the example above, the mapping table shows that the ARM object `structure_enclosure_element` maps to the AP-created AIM object `structure_enclosure_element`, which is a subtype of `product_definition` from the IRs. The fact that the `structure_enclosure_element` is a wall is represented by the name of a `product_category`.

The mapping table is complex and a challenge to understand, but is the critical linchpin within the AP.

The conformance requirements specify the fundamental characteristics and conformance classes for compliant implementations of the AP. A conformance class defines the subset of the ARM and the corresponding subset of the AIM required for useful and compliant implementations of the AP.

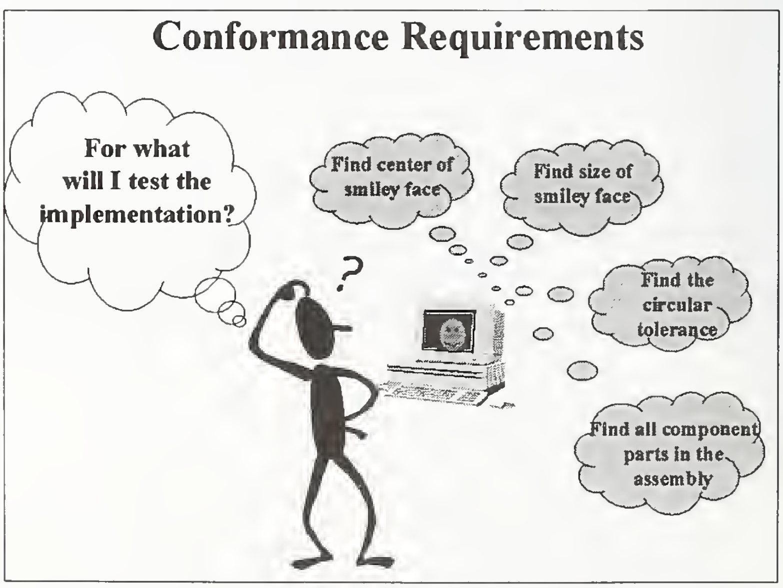


Figure 7-6: Application Protocol Conformance Requirements

The complete content requirements of an AP are provided below in Table 7-1.

Foreword
Introduction
1 Scope
2 Normative references
3 Definitions and abbreviations
4 Information requirements
4.1 Units of functionality
4.2 Application objects
4.3 Application assertions
5 Application interpreted model
5.1 Mapping table
5.2 AIM EXPRESS short listing
6 Conformance requirements
Annexes
A AIM EXPRESS expanded listing
B AIM short names
C Implementation method specific requirements
D Protocol Implementation Conformance Statement (PICS) proforma
E Information object registration
F Application activity model
F.1 Application activity model definitions and abbreviations
F.2 Application activity model diagrams
G Application reference model
H AIM EXPRESS-G
J AIM EXPRESS listing
K Application protocol implementation and usage guide
L Technical discussions
M Bibliography
Index
Figures
Tables

Table 7-1: Contents of a STEP application protocol

7.4 DEVELOPING AN APPLICATION PROTOCOL

The first phase of AP development is defining industries' priorities and needs for reliable information exchange standards, and then establishing international participation in developing the AP to meet these needs. SC4 works to ensure that experts from all intended industrial users participate in this task. With the industry priorities and needs documented, the definition of the scope and information requirements begins with the formulation of a concise statement of the application context and functional requirements for the AP. This statement defines the product data application(s) targeted for the AP and the intended use of the product data within the application(s). The detailed scoping and information requirements definition proceeds from this statement.

Scope definition is refined via an application activity model (AAM). The AAM describes the use of the product data within the application domain with a process modeling technique such as IDEF0. During this analysis example, products and usage scenarios from the application domain are documented. These usage examples are extremely valuable in defining information exchange requirements and in subsequent validation testing of the ARM and the AIM.

When the detailed scope and general information requirements have been defined, the information domain of the AP is described by the use of the application reference model (ARM). The ARM is developed using a standard information modeling technique: IDEF1X or EXPRESS. Each application information requirement deemed within scope must be expressed in the ARM. The ARM is sufficiently detailed to describe fully the information requirements of the application domain.

A basic mechanism to modularize the scope of an AP into manageable components is to define units of functionality (UoFs) within the context of the ARM. A UoF is a collection of entities, attributes, and relationships that conveys one or more well-defined concepts within the context of the ARM. UoFs are used to organize the ARM into easily understood and logical groups of concepts and provide a basis for defining the conformance classes for the AP. NIST has developed a web-based, internet-accessible tool for browsing existing UoFs. The site provides all UoFs associated with each AP, the relationship of a UoF to other UoFs, and the UoF definition [142]. The site presents requirements of APs such that new AP developers may find related requirements in existing APs. Reusing these requirements facilitates harmonization of APs at the requirements level.

The AIM is developed by selecting and constraining constructs from the integrated resources to convey the concepts and information requirements of the ARM. The process of developing the AIM includes ensuring consistency of STEP data representations across APs and the reuse of the same constructs for representing the same information in different APs. Developing and validating a STEP AP is an iterative process of progressive detailing and validation testing.

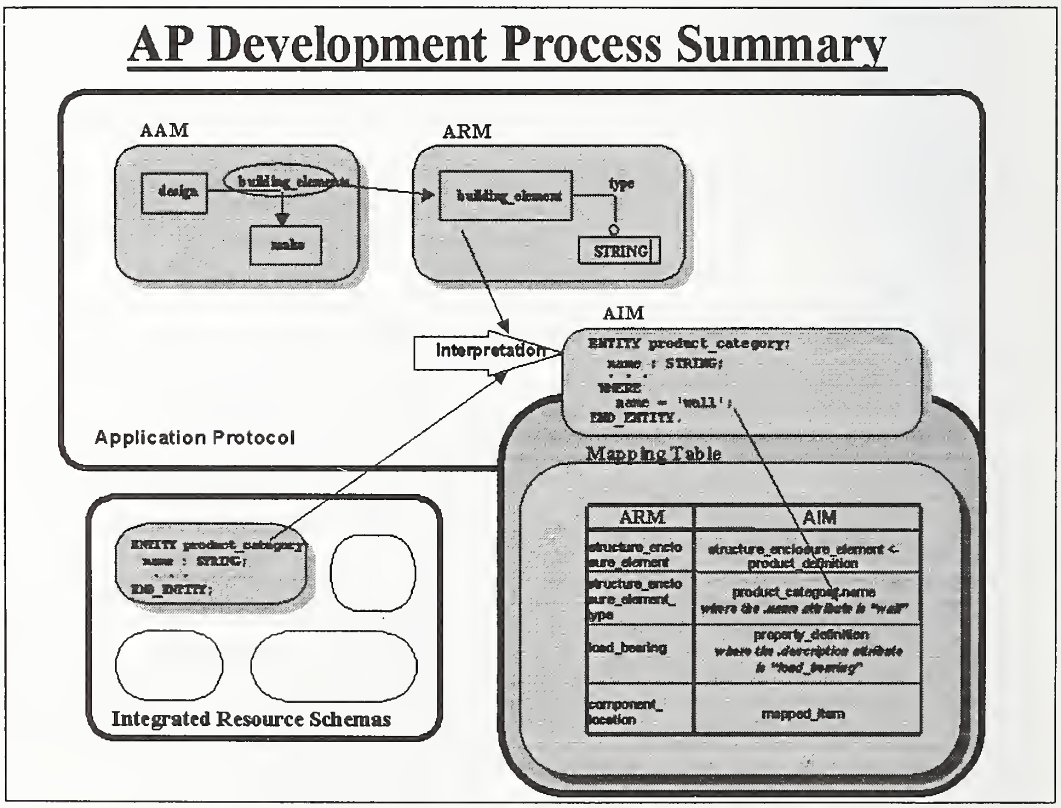


Figure 7-7: Application Protocol Development Process

Figure 7-7 provides a summary of the AP development process. Briefly, the information flows in an AAM lead to information objects in the ARM. The interpretation process identifies IR objects that correspond to ARM requirements. An express AIM is created that includes a subset of the IRs and specializations of those constructs.

The correlation between the ARM and the AIM is documented in a mapping table. The mapping table provides guidance on the use of AIM constructs within that application.

7.5 PLANNING AND MANAGING AP PROJECTS

From 1990 to 1998, SC4 authorized the start of thirty-two AP projects. During this time, SC4 established procedures to assess industry requirements and to promote timely completion of APs as International Standards. These efforts included the use of AP planning projects to facilitate the definition of suites of APs to meet the needs of industry sectors, e.g., shipbuilding and process plants. There were also efforts to collect, synthesize, and generalize industry requirements for product data exchange and sharing. Additionally, industry was encouraged to define industry-wide AAMs and information technology roadmaps to aid in the future definition of industry needs and priorities.

The aerospace industry has taken a general approach to using STEP in using application protocols that are generic in nature, e.g., configuration management. The automotive, shipbuilding, and process plant industries have developed specific APs for their industries' needs. For example, the scope of ISO 10303-214 is "core data for automotive mechanical design processes;" it will be used, at a minimum, by automotive companies in the U.S., Europe, and Japan.

The shipbuilding industry is building a suite of five integrated APs. The chemical, process plant, and architectural engineering and construction (AEC) industries are also developing APs that are focused on their particular industries.

Of the thirty-two AP projects started in this same period, five have become international standards. Table 7-2 lists all the active AP work items in SC4 at the time of this publication. (*International Standards)

Application Protocol Number and Title	
10303-201*	Explicit draughting
10303-202*	Associative draughting
10303-203*	Configuration controlled design
10303-204	Mechanical design using boundary representation
10303-205	Mechanical design using surface representation
10303-207*	Sheet metal die planning and design
10303-208	Life cycle management 10303- Change process
10303-209	Composite and metallic structural analysis and related design
10303-210	Electronic assembly, interconnect, and packaging design
10303-212	Electrotechnical design and installation
10303-213	Numerical control process plans for machined parts
10303-214	Core data for automotive mechanical design processes
10303-215	Ship arrangement
10303-216	Ship moulded forms
10303-217	Ship piping
10303-218	Ship structures
10303-221	Functional data and their schematic representation for process plant
10303-223	Exchange of design and manufacturing product information for casting parts
10303-224*	Mechanical product definition for process plans using machining features
10303-225	Building elements using explicit shape representation
10303-226	Ship mechanical systems
10303-227	Plant spatial configuration

Application Protocol Number and Title	
10303-230	Building structural frame: Steelwork
10303-231	Process engineering data: Process design and process specification of major
10303-232	Technical data packaging core information and exchange

Table 7-2: Current Application Protocols

Many attempts have been made to reduce the huge investment of resources and time necessary to produce an AP. Attempts to reduce the requirements, alter the methods, or build tools to automate AP development have, at best, made only small dents in these intensive undertakings. This current approach to development raises continued debate on a couple of issues:

- Should application protocols be standardized or should only the infrastructure parts of ISO 10303 (10s, 20, 30s, and 40s series of parts) be standardized?

- Should application protocols be harmonized only within industry sectors or continue to be integrated across industry sectors?

The automotive and aerospace industries were early adopters of ISO 10303 and participated in the development and deployment of the first STEP APs. Through the cooperation of their international industry consortia, these industries built a core competence in this technology. Both industrial sectors continue to work to ensure the utility and reliability of the supporting APs to be used in their supply and delivery chains. With the introduction of the AP methodology, many of the industry sponsors for STEP shifted their attention and resources to AP projects that met their specific information requirements. This migration of resources, mentioned in Chapter 3, dampened the efforts on delivering the common integrated resources and potentially delayed the delivery of improved Application Interpreted Constructs (AICs).

As more industries investigate their needs for information exchange and sharing standards, SC4 is confronted with an expanding set of industry expectations and requests for additional capabilities and mechanisms for standardizing industry semantics for many types of communications. These requests range from archival of design intent to data warehousing using industry classifications of reference data. Some fragmentation has occurred because of a focus on isolated industrial solutions. These requests have highlighted some of the limitations of ISO 10303 and the AP development process. Such highlights contribute to the need for an SC4 architecture of standards and strategic plan for delivering the needed standards.

7.6 CONCLUSION

SC4, through the STEP project, pioneered the combined use of process modeling, information modeling, mapping between data models, and conformance testing across different implementation paradigms. Each of these methodologies, and the corresponding software tools, has matured through the AP delivery process during the past eight years. The weakest aspects of the existing process today are:

- Information model mapping notations and tools.
- Inefficiency and speed of the standards development process.
- Inadequate and inconsistent means to scope an individual AP.
- Small amount of industrial testing of the draft standards.
- Lack of tools to facilitate implementation of the APs.
- Difficulty in changing or enhancing a published standard.

The mapping table in an AP is its most complex and overwhelming component. SC4 has improved some of the utility of the mapping table. Yet, until the mapping table is computer interpretable, it will continue to be a major barrier to the understanding and use of STEP APs.

The original AP development process provided a well-defined means of collecting industrial requirements and input for prioritizing the work to complete the functionality and semantic capabilities of STEP. The SC4 Project Management Advisory Group (PMAG) initially promoted this aspect of the AP development process, and NIST investigated potential tools for improving the collection and synthesis of STEP requirements.

Unfortunately, at the same time that NIST, with CALS support, was developing a prototype STEP Requirements Management System, the PMAG and SC4 decided to disband WG4 (see Chapter 9) and not reassign many of WG4 core responsibilities. This resulted in removing from the prescribed SC4 AP development process any analysis of cross industry requirements and the emphasis on aligning AIM structures. With this change of emphasis, there was no longer any centralized mandate to ensure high value commonality across the APs. Although this change was motivated by the interest to remove the perceived "SC4/WG4 bottleneck," it disabled a fundamental objective of the AP development process and promotes fragmented solutions.

Some industry projects are supporting additional testing of the draft standards and developing tools to facilitate the implementation of the APs. Improvements in these areas and the continued promotion by industry leaders for the use of STEP APs will help to ensure continued, broad adoption by industry despite fragmentation. Implementing APs still promises the delivery of significant business benefits from this technology.

CHAPTER 8

CONFORMANCE AND INTEROPERABILITY TESTING

8.1 INTRODUCTION

Modern, information-based engineering systems are composed of various components that share product data. These components are generally software entities, which are designed to address certain functions such as design, analysis, manufacturing, and data management. Many different commercial implementations of any specific component may exist that will satisfy the functional requirements established by the user. Users must weigh an application's features and function, along with its capability to share and exchange data.

To share data, components within a system must adhere to a common interface specification or standard. The fact that functionally-equivalent implementations must necessarily compete, and the fact that they must adhere to a common interface, creates an impasse. Applications must be different to compete, but they must be the same (share the same data structure) to share data. Vendors often provide special features within their products[28] as one way of allowing their products to be differentiated from the products of other vendors. Standards developers compensate for this by providing very strict and detailed definitions of conformance.

Applications usually exhibit non-conformance in one of two ways. The first occurs when certain features fail to conform to the behavior defined in the standard, or the feature is not present in the implementation. The vendor may have misinterpreted the specification, or failed to verify fully the behavior of the implementation, resulting in failure to conform to the standard. The second occurs when the application circumvents the interface specification to permit function or features not addressed in the standard. An example of this would be a vendor defining more attributes in an interface than were included in the standard. Both types of non-conformance can cause interoperability and portability problems that will result in significant development and maintenance costs.

Application vendors will sometimes use the term "compliant" when describing their product. This usually means the vendor feels it has implemented a standard sufficiently and possibly that it has performed some testing to determine if the product is conforming to the requirements of the standard. Conformance testing is a formal means of verifying compliance to the standard and implies that the test methods and requirements are under the control of an organization other than the vendor developing the product.

Conformance testing of implementations implies a formal approach to developing test methods and test requirements that provide the framework for verifying a vendor's claims of compliance with a standard. Different paths have been taken in developing formal conformance testing. The most comprehensive approach is to develop conformance test requirements in conjunction with the creation of a standard. In this approach, a working group within the standards body is formed to develop the conformance test methodology in conjunction with the standard. The ability to leverage the gathered expertise to resolve ambiguities related to test requirements in an open forum results in a concise and accurate set of conformance test standards that are controlled by the standards body. Other than ISO 10303, historical examples of this approach are: the American National Standards Institute (ANSI) X3T9.5 Fibre Distributed Data Interface (FDDI) Conformance Test Working Group, IEEE 896.4 Futurebus+ Conformance Test Working Group, and IEEE 1003.3 POSIX Conformance Test Working Group [143]. These testing programs have been active now for more than a decade and many of the desired results from these activities have been achieved.

---

[28] The use of the word "product" is used interchangeably with the word "implementation" throughout this chapter. Both are intended to mean commercially available software based on a standard.

Early in ISO 10303 development, NIST, CALS, CADDETC[29] in the United Kingdom, and key prospective STEP users, who desired the acceleration of STEP products to market, worked to initiate activities to develop better methods and tools for conformance and interoperability testing. NIST has a long history of working on testing product data exchange standards. In the early 1980s, NIST received funding from the CALS Program to develop testing methods for IGES subsets. As part of the National PDES Testbed activities, CALS funded NIST's efforts to perform validation testing, first for the PDES, Inc. CDIMS, and then for the early versions of ISO 10303. In the early 1990s the Navy Mantech program funded a joint effort by NIST and the Industrial Technology Institute (ITI), Michigan, to develop conformance testing methods. This funding was supplemented with CALS funding. For the last few years, NIST has continued to support the testing efforts using Department of Commerce funding. Experience with previous standards had demonstrated the importance of defining concise, unambiguous conformance requirements for STEP in order to prevent the development of conformant, yet incompatible, STEP implementations. Lessons learned from previous standards, such as ISO 9646 [144], had also shown conformance testing requirements and test methodologies must be available at the time the standard is published. This chapter discusses the purpose of testing, the various types of testing, benefits of testing, and describes developing conformance testing methodologies for ISO 10303 APs.

8.2 TESTING

Developing and integrating products based on complex standards is extremely difficult. One effective method for accelerating the integration process is conformance testing. Testing a product to an established standard using agreed-upon references can help determine if the product will be able to interact with other products that adhere to the same standard. Various types of testing methodologies are used during product development and deployment.

Validation testing – the assessment of the underlying specification to which products will be developed. Validation testing attempts to evaluate the completeness, correctness, and consistency of a data model to be used for a standard.

Conformance testing - the testing of a candidate product for the existence of specific characteristics required by a standard in order to determine the extent to which that product is a conforming implementation.

Interoperability testing – the assessment of a product to determine if it will exchange and share information (interoperate) with another product implementing the same specification.

Performance testing – the assessment of the performance characteristics of a product such as throughput and response time under various conditions.

Robustness testing – the assessment of a product to determine how well it performs when supplied data that is difficult to process, such as, extremely large data sets or data which contain errors.

Acceptance testing – the process of determining whether a product satisfies predefined acceptance criteria. Acceptance testing is a combination of other types of tests to demonstrate the product meets user requirements.

The two primary approaches for achieving general system integration are conformance testing and interoperability testing.

[29] CAD-CAM Data Exchange Technical Centre, at University of Leeds, United Kingdom

8.3 CONFORMANCE TESTING VERSUS INTEROPERABILITY TESTING

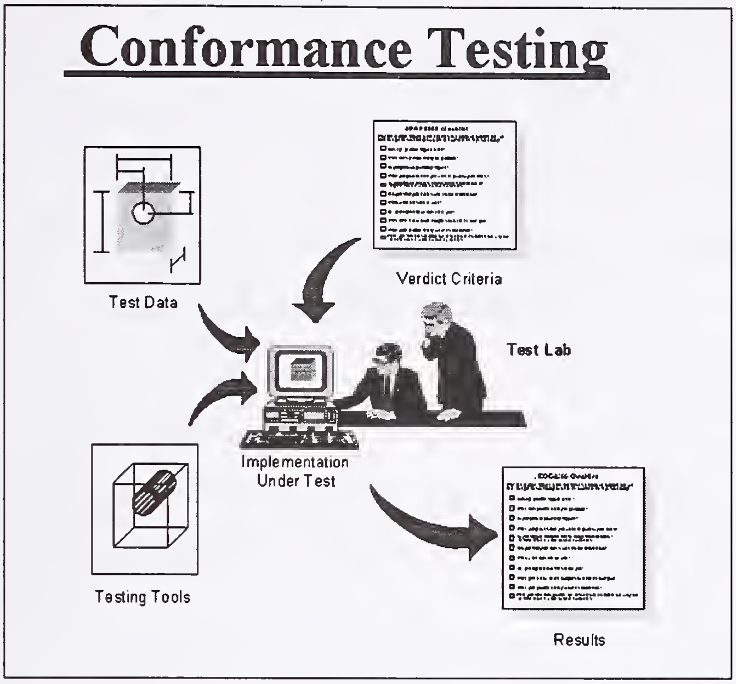


Figure 8-1: Conformance Testing

Conformance means meeting the specified requirements. In conformance testing (Figure 8-1), an implementation is tested using specified test cases to check that it meets the specified requirements with respect to the options that it is said to support. The requirements to be met for conformance should, ideally, also be those that are necessary to support interoperability. Conformance testing, therefore, provides a high level of confidence, but <u>not a guarantee</u> that systems will interoperate. Conformance testing *does* <u>provide</u> a basis for determining whether products implementing ISO 10303 will interoperate.

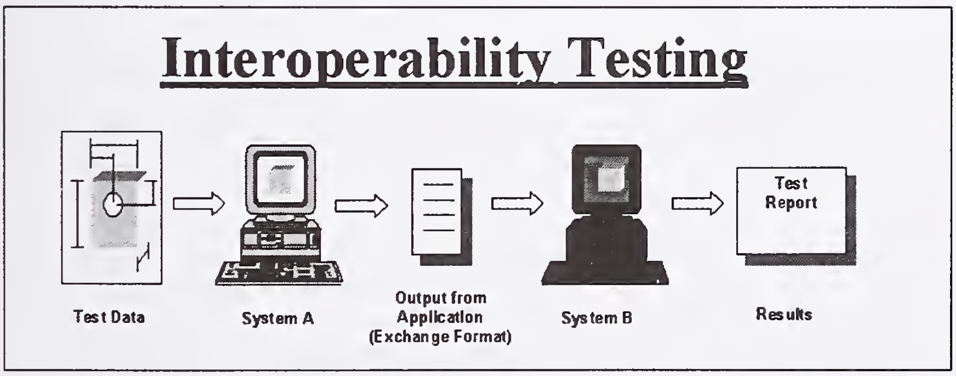


Figure 8-2: Interoperability Testing

"Interoperating" means working together. In the context of STEP, it means exchanging and understanding information between two different systems implementing the same AP. Interoperability testing (Figure 8-2) is used to determine whether an implementation can be made to function effectively with another implementation. The advantages of interoperability testing are that it requires fewer test cases and the results from such testing are direct rather than inferred. A successful interoperability test indicates two implementations <u>will</u> interoperate, while a successful conformance test indicates two implementations are only <u>likely</u> to interoperate. In this sense, interoperability testing, from the viewpoint of the user, is more effective than conformance testing; however, when testing one implementation to determine if it will work with many others, there is a cost trade-off between testing once in a rather thorough way and testing many times in a simpler way. Given N implementations, 2*N conformance tests are required because each implementation is tested once for input (post-processing) and once for output (preprocessing). For the same N implementations, N^2 interoperability tests are required (Figure 8-3). As the number of systems to be tested increases, exhaustive interoperability testing becomes less practical due to the large number of individual tests required. Thus, ensuring interoperability has an inherited expense proportional to the thoroughness in which it is carried out.

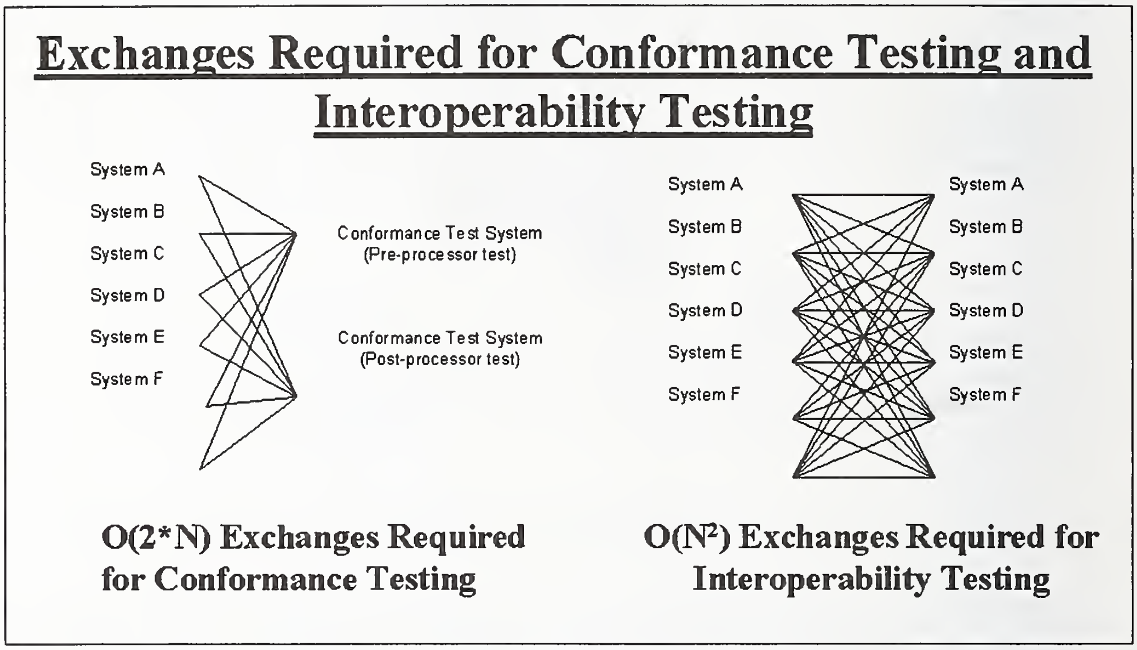


Figure 8-3: Required Exchanges

Conformance testing provides some important advantages over interoperability testing as defined above. Conformance testing first and foremost pays strict attention to the standard. This means that conformance testing, more so than interoperability testing, ensures that all of the requirements of the standard have been met in an implementation. Drawing an analogy to electrical circuits, conformance testing serves as a reference, a grounding point, for all standards-based implementations to share. Interoperability testing without conformance testing usually results in pair-wise implementation drift: an implementation ends up having to be reconfigured in some manner to compensate for another implementation's slightly nonconforming behavior. Conformance testing also makes it easier to localize a problem. Since there is only one implementation being tested there is no ambiguity as to which system is at fault when a test fails.

Conversely, interoperability testing has certain advantages over conformance testing. Since it is not constrained to the requirements of a standard, interoperability testing can look at other factors that are of interest to the user. It can be used to focus on factors, which may prevent interoperability that will not be addressed by the standard. One example of this is CAD system accuracy mismatch where CAD vendors use different granularity for tolerance specification. Another example is different design practices. With interoperability testing, the user can focus on the areas that are important to their operations, which are not addressed by, and are independent of, the standard. Test data specific to the domain of interest can be used. This provides additional assurance that the systems will be able to exchange information during actual production use.

Approach	Conformance Testing	Interoperability Testing
Validates an implementation against:		
explicit requirements of the standard	✓	
user-driven requirements		✓
Tests against trusted reference system	✓	
Identifies interoperability issues		
within the scope of the standard	✓	✓
outside the scope of the standard		✓
beyond implementations being tested		
Identifies IUT at fault	✓	
Broad coverage of standard	✓	
Number of tests for N systems	O(2*N)	$O(N^2)$
Formal	✓	

Table 8-1: Conformance Testing versus Interoperability Testing

Table 8-1 summarizes the coverage of each approach. Ideally, both kinds of testing should be performed, but the costs of doing so can be prohibitive. One approach combines the advantages of both testing methods. Such an approach is possible by combining some of the discipline and tools used in conformance testing to the interoperability testing process.

8.4 DEVELOPING CONFORMANCE TESTING METHODOLOGIES FOR ISO 10303

As mentioned in the prior chapter, the implementable parts of ISO 10303 are the application protocols (AP). APs also define the conformance requirements, grouped into structures called conformance classes. A conformance class is a specific subset of a complete AP that defines a valid implementation. A conforming implementation must support all requirements within a given conformance class. Vendors may not select a subset of the concepts from a conformance class to support and still claim conformance.

Shown in Chapter 7's Table 7-1, a STEP AP has two clauses, which are especially critical for testing a STEP implementation:

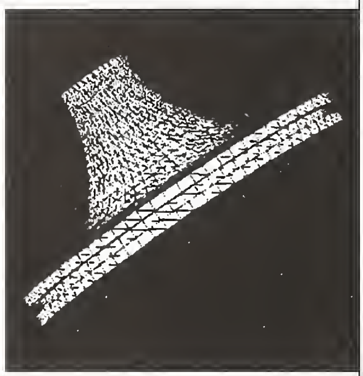
Clause 4: Information Requirements – The specific definitions of the AP semantic elements are given as a set of Application Elements, consisting of objects, attributes, and assertions between the objects that are derived from an Application Reference Model (ARM) within the Scope of the AP.

Clause 5: Application Interpreted Model – The Application Interpreted Model (AIM) provides computer-sensible language definitions of entities, data types, and constraints in an EXPRESS [145] schema. Clause 5 also defines the Mapping Table - a specification of the precise encoding of each Application Element of Clause 4 in terms of the constructs defined in the AIM.

As mentioned before, each 10303 AP has an associated abstract test suite (ATS). An ATS contains the set of abstract test cases (ATCs) for an AP to support the conformance requirements. Each ATC provides an implementation-independent specification of the actions required to evaluate part of one or more conformance requirements. Each AP contains a normative reference to the corresponding ATS.

Each conformance requirement corresponds to one or more ATCs, designed to satisfy one or more test purposes. Test purposes are singular, precise descriptions of test objectives, e.g., "test the generation of a curve as a composite curve with senses defined." Sufficient test purposes must be written to provide adequate coverage of the entities in the AP. For each ATC, verdict criteria are generated from the conformance requirements to allow a testing laboratory to assess the conformance of an implementation with respect to that test case. When a conformance test based on an ATC is conducted, the resulting verdict indicates whether the implementation meets one or more conformance requirements [146]. Test cases are defined using a formal language specified in ISO 10303-34 [147]. Figure 8-4 is an example test case for ISO 10303-203 [148].

ISO 10303-203 Test Case



Test Case: AB_Revolved_Object_With_Void

Test Summary:
Test the processing of information elements as defined in the UoF advanced_boundary_representation. Specifically, the postprocessor input contains instances of adv_brep_shape_rep, brep_with_voids and elements used in their consruction (CC6).

Input Specification:
(Model design, application elements and values)

Verdict Criteria:
General verdict criteria apply: gvc1, gvc2, and gvc3

VC1:The model realized in the IUT must generally correspond in shape to the model represented

VC2::This model can be created by revolving a composite curve 360 degrees about an axis. The length of the axis is 0.49 m.

VC3 :This model contains a void. The volume of this model, excluding the void, is 0.4330 cu.m.

Figure 8-4: ISO 10303-203 Test Case

The importance of incorporating conformance testing into ISO 10303 was recognized early in developing STEP. As mentioned in Chapter 4, NIST played a strong leading role in raising SC4 consciousness to appreciate the value of conformance testing. The 30-series parts, "Conformance testing and methodology and framework," specify the requirements for, and provide guidance on, procedures to be followed in conformance testing for ISO 10303. These parts provide information necessary to meet the following objectives: confidence in test results, comparability between test results, and communication between parties responsible for testing. The 30-series parts define the abstract test suite requirements for application protocols, the abstract test methods for ISO 10303 implementation methods, common terms and concepts, and the conformance assessment process carried out by a testing laboratory.

ISO 10303 30-series parts:

- 31: Conformance testing and methodology framework: General concepts.
- 32: Conformance testing and methodology framework: Requirements on testing laboratories and clients.
- 34: Conformance testing and methodology framework: Abstract test methods for application protocol implementations.
- 35: Conformance testing and methodology framework: Standard Data Access Interface.

IDEF0 [149] was used to document the conformance testing procedures for ISO 10303. Figure 8-5 shows the overview of the conformance assessment process. The decomposition of this model provides the basis for the test methods in 10303-34. In order to initiate the conformance assessment process (undertake a test), a test system must be created. Although the test system itself is not standardized, it is based on the requirements in ISO 10303.

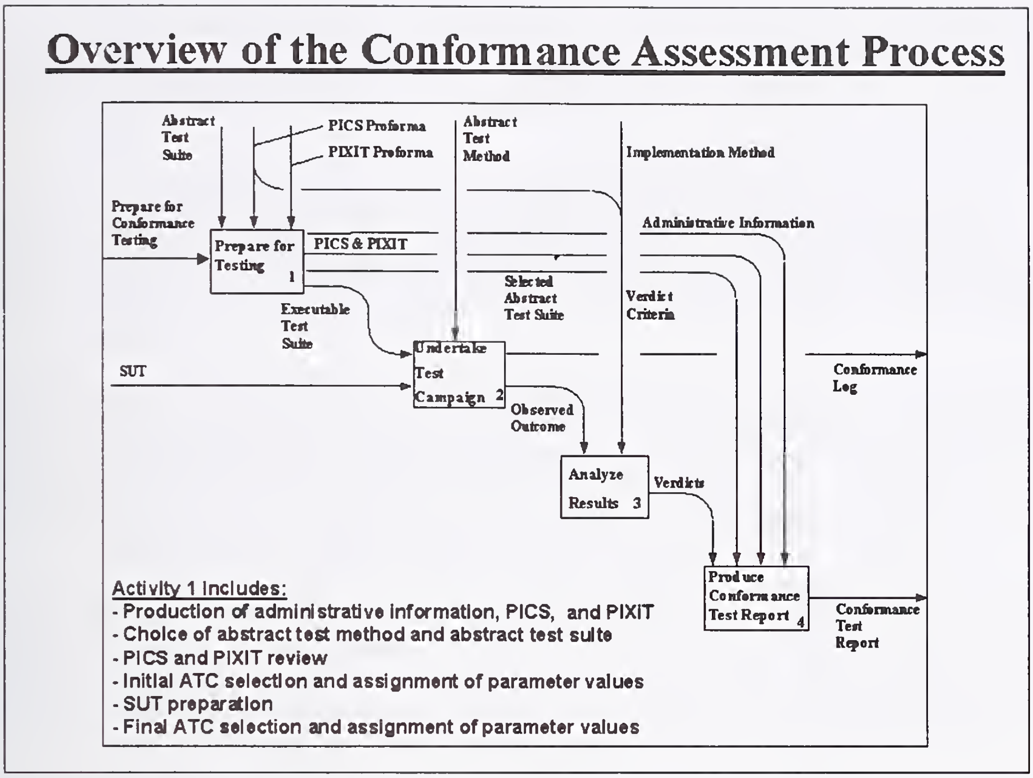


Figure 8-5: Conformance Assessment Process

Elements of the executable test system are derived from the standards documents. An executable test suite is derived directly from the ATS. An overall framework for conformance testing and certification is provided in 10303-31 [150]. Guidance for developing forms for gathering extra information about the system being tested (PIXIT proforma) is given in 10303-32 [151]. Guidance for developing detailed procedures manuals is provided in 10303-32 and -34. Software tools for assessing conformance are based on requirements in 10303-34.

8.5 STEP CONFORMANCE REQUIREMENTS

ISO 10303 APs require all entities of a conformance class to be implemented in order to achieve conformance. This explicit implementation requirement is necessary to avoid the confusion that can result from specifications that have mandatory elements and optional elements. In the past, standards with optional elements have resulted in "flavored" implementations - a flavor being the set of optional elements used by a specific application and the interpretation of those elements. Applications that conform to the same specification, but use different sets of optional elements may not be compatible. The IGES neutral format specification and the RS274 machine tool control language specification [152] are examples of standards that include optional elements and that have resulted in many incompatible commercial implementations. To address this situation, STEP developers included strict conformance requirements, defined test purposes, added a requirement to include an implementation conformance statement, and standardized the ATSs for each AP. STEP developers standardized the ATSs in order to alleviate the informal development of multiple abstract test suites by various testing groups [153].

8.6 CONFORMANCE TESTING STEP IMPLEMENTATIONS

An implementation of a STEP AP is either a *preprocessor* or a *postprocessor*. A preprocessor is an implementation that generates an ISO 10303 AP information model or exchange structure. A postprocessor is an implementation that interprets an ISO 10303 AP information model or exchange structure. In both cases, the function (behavior) of the Implementation Under Test (IUT) is the same: translate the information from the given input format to the prescribed output format. Essential to this translation is the preservation of the semantic content of the information that is contained in the model. Implementations of a STEP AP may affect this translation using either of two defined implementation methods described in Chapter 6: *file exchange* using the exchange structure defined in ISO 10303-21 [154], or standard access to a *shared database* via the Standard Data Access Interface (SDAI) defined in ISO 10303-22 [155].

The general conformance testing process has been applied directly to testing implementations of an ISO 10303 AP [156]. The Implementation Under Test (IUT) supplies an instance of the AIM schema defined by the STEP AP in a given input format and is expected to translate it into an appropriate output format. The testing inputs are driven by the AP specification, controlled by the testing system, and directly related to the analysis applied to the outputs produced.

For a preprocessor, the input provides information represented in a data format native to the originating system. Such input is a subset of the information requirements of an AP. While the semantics of the input is well defined in STEP, the specific format of this input is not. For conformance testing purposes, this format has been defined as a form of human readable text and graphics termed "hardcopy." The action of translation for a preprocessor produces an output instance of an EXPRESS AIM schema. The format of this output instance model is either a STEP exchange structure (for file exchange) or a series of SDAI calls (for a database implementation).

For a postprocessor, the situation is reversed. The input is an instance of an EXPRESS AIM schema, in the format of a STEP exchange structure (file exchange) or a series of SDAI calls (database). Translation for a postprocessor produces an output with the specific format undefined by STEP. For conformance testing purposes, postprocessor output is a series of responses to queries about the semantics of the information contained in the input model instance. Though a postprocessor may provide hardcopy output, this is not assumed or required.

Analysis applied to the output produced by an IUT assesses capability in three areas: syntax, structure, and semantics. In STEP, testing syntax and structure analysis applies only to preprocessor testing while semantic analysis applies to both preprocessor and postprocessor testing.

Syntax analysis – applies only to the output exchange structure of a preprocessor. Syntax analysis checks that all the requirements of the application's implementation method are satisfied, either the file format as prescribed in ISO 10303-21 or the standard access methods as defined in ISO 10303-22.

Structure analysis – ensures that the data model represented in a preprocessor output exchange structure satisfies all structural requirements of the AIM EXPRESS schema defined by the AP. This includes the verification of all data types as well as all locally and globally defined constraints.

Semantic analysis – verifies that the semantics defined by the information requirements of the application protocol of interest are conveyed accurately in the observed output. Semantic analysis applies to both preprocessors and postprocessors.

The conformance test provides input data in one format and verifies the correctness of the output produced by the IUT in another format. An implementation successfully passes conformance testing when the syntax and structure of the output conforms to the requirements of the standard, and when the semantic content of the output is equivalent to that of the input.

8.7 INTEROPERABILTY TESTING OF ISO 10303 IMPLEMENTATIONS

Interoperability testing is not part of the ISO 10303 series of parts, a normative requirement imposed by any of the ISO 10303 parts, nor an official activity within the SC4 community. It *is* viewed as a critical contribution to the success of STEP adoption. Successful information exchange among implementations offers a higher confidence level to users for STEP adoption.

Interoperability test typically produces an output exchange structure from one IUT based on some internal model with known semantics, and then uses this exchange structure as the input to another IUT. The correctness of the internal model within the second IUT is then verified. Implementations successfully pass interoperability testing when the internal representation of the first IUT is equivalent semantically to the internal representation of the second IUT (Figure 8-2).

As stated previously, interoperability testing is not constrained to the requirements of a standard. Consequently, interoperability testing can be used to examine factors that are of primary interest to the user. Successful interoperability testing is based on fundamental design of experiments. Interoperability test requirements and procedures must be understood completely prior to the test. The following test planning elements are critical for successful test implementation.

Exchange scenario – How will data be used?

Exchange metrics – How is success measured? What constitutes a "successful data exchange?"

Exchange procedures – How will exchanges be accomplished? By whom? How many?

Controls – What are the possible sources of exchange errors? How will these be isolated?

Results analysis – how will results be presented?

8.7.1 Exchange Scenario

The exchange scenario is the assumed, ideal, or desired paradigm for the specific data exchange to be tested. It defines the scope and provides the basis for developing requirements and test procedures. The exchange scenario is used to identify the purpose of data exchange, define the qualities of successful data exchange, identify performance parameters, select appropriate test parts, and identify metrics to measure exchange. An example of an exchange scenario from the Automotive Industry Action Group (AIAG) AutoSTEP Interoperability Test Program [157] is shown in Figure 8-6.

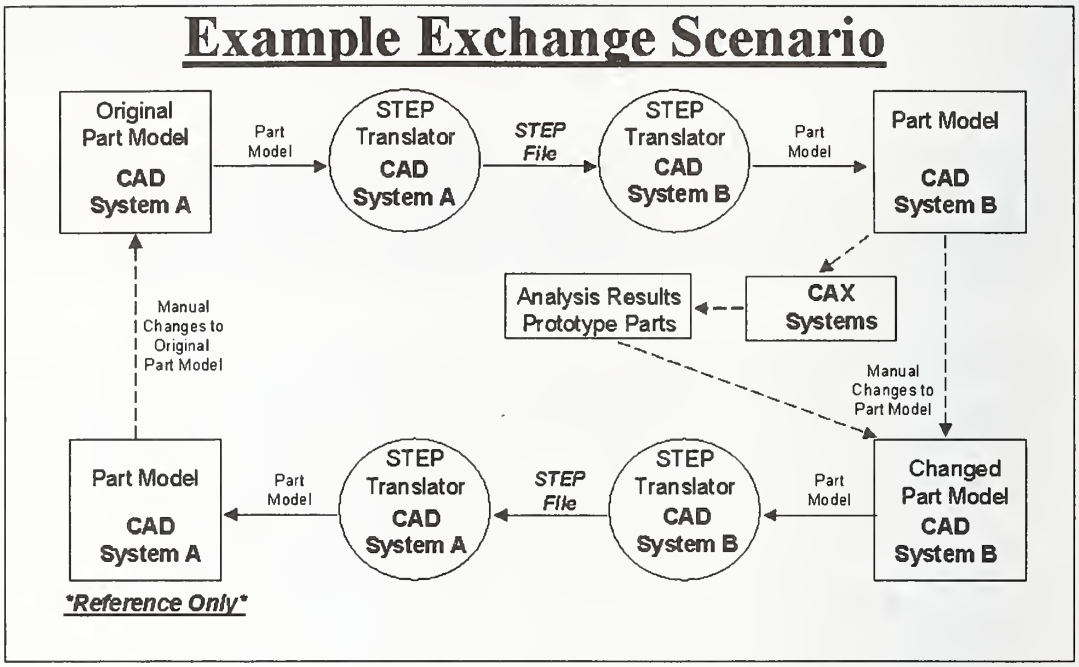


Figure 8-6: Example Exchange Scenario

Model data in the exchange process transitions through stages, or states. At each state, a unique set of metrics is gathered to record information about the model at that state. The values of the metrics are checked to determine if any errors or abnormalities have been introduced into the model during translation. Test metrics must:

- Provide information relevant to test requirements.
- Have equivalent meaning in all systems to be tested.
- Be objective (not subject to interpretation).
- Be reasonably easy to collect.

There are two fundamental types of metrics: simple metrics and process metrics. Simple metrics are parameters that may be calculated directly from the model. They are used to check for actual error conditions in the model. Process metrics are metrics that are not computed directly from the model, but are determined by comparing the simple metrics of the model at different states in the exchange process. Examples of process metrics include change in mass properties between an exported and imported model, change in number of entities, or change in file size. A variety of tests and metrics are used to determine whether translations are successful. Metrics are selected based on application requirements. A list of typical metrics and metric-related activities of concern to the user is given here:

Communications
- file size (file growth)
- translation time

Diagnostics
- system errors
- model validation (system dependant)

Geometry
- closed shell
- surface area

- volume
- mass properties (e.g., centroids)
- number of surfaces
- number of faces
- number of solids in assembly
- specific measurement (e.g., point to curve)
- colors
- layers
- gaps
- extraneous feature creation (e.g., sliver faces, micro edges, etc.)
- model complexity

Non-Geometry
- relationships
- text strings
- variables

Functionality
- visual inspection
- manipulate model
- save in native format
- modify model
- target application functionality

In many cases, interoperability tests occur in a distributed environment with many different system operators involved. A data exchange failure may have any number of possible sources, many having nothing to do with the systems being tested. It is important to validate intermediate data in order to minimize the number of variables and isolate the test systems. Test procedures must be established in order to ensure that all test data is recorded properly.

8.7.2 Analysis of Results -- Conformance Testing Tools

To understand how testing is enhanced by conformance testing tools, it is important to understand the nature of the tools that have been developed. The list below describes tools that were developed by NIST and the Industrial Technology Institute (ITI)[30], Ann Arbor, MI.

The Testing Harness - The process of organizing test suites, executing the tests, analyzing the outputs, compiling the results into reports, and yielding an overall verdict can be labor intensive. A reusable testing harness has been developed to handle these administrative tasks. The harness can be adapted to any standard by plugging in the appropriate standard-specific test suites and analysis tools.

The Test Purpose Generator - Many of the test purposes required in a test suite can be generated automatically from the schema. This tool reads the schema defined in a 10303 AP and generates a corresponding list of AIM-derived test purposes. This tool also reads the ARM Application Elements (AEs) and Assertions and generates the corresponding list of AE-derived test purposes.

[30] In 1998, ITI changed its name to ERIM: Environmental Research Institute of Michigan

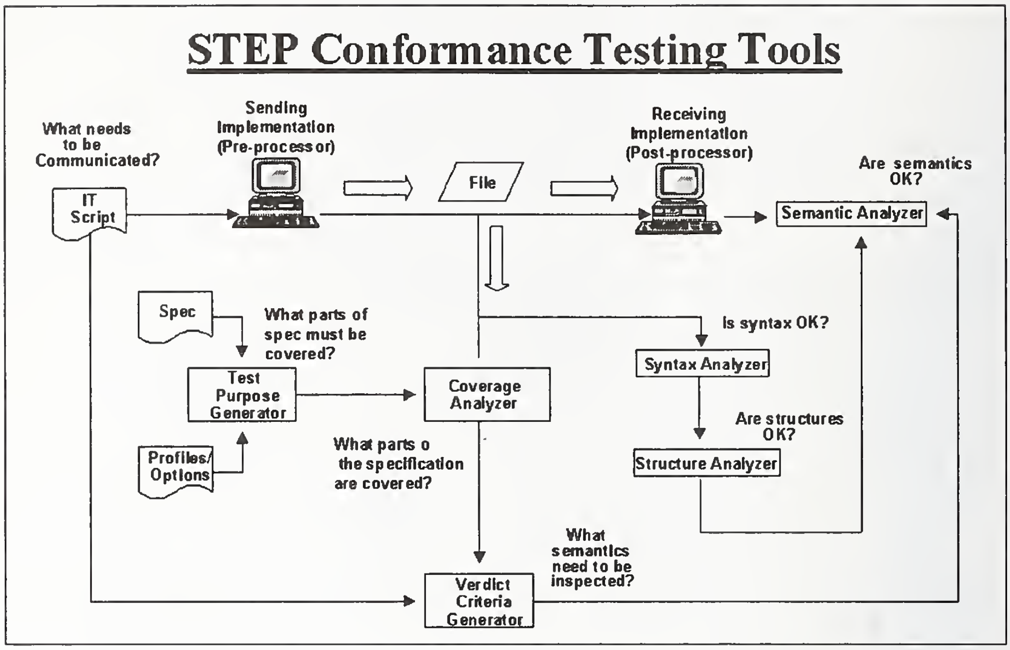


Figure 8-7: STEP Conformance Testing Tools

The Coverage Analyzer - The coverage analysis tool is used to compute the percentage of coverage of a test data set against a specification. It evaluates the degree of coverage or completeness of a test suite against the test purposes. The percentage of all possible test purposes that are satisfied by the input data provides a measure of test suite coverage over the stated testing objectives. The Coverage Analyzer can be used to:

- Compute the coverage of a test suite against the AIM, ARM, and other test purposes in the test suite development process.
- Estimate the degree of interoperability of two STEP implementations by computing the coverage attained by the files exchanged between the systems against the test purposes.
- Estimate the added value of new test data to a test suite by computing the degree of overlap with existing test data.

The Verdict Criteria Generator - The verdict criteria generator is able to read the input specifications of a STEP test suite and generate meaningful verdict criteria from that data. Most test suites include a large percentage of overlapping data. Consequently, the same test purpose may be covered by multiple test cases.

The STEP File Checker - This tool checks the format of STEP exchange files (ISO 10303-21) for proper syntax and structure. It ensures that all the rules defined in an AP are maintained in a STEP exchange file. These rules include not only the simple data constraints in a STEP schema, but also the more complex geometry and topology rules applied to the geometric models.

The ARM/AIM Browser and Editor - This tool provides the capability to work with STEP data from the perspective of both the ARM information requirements and the AIM EXPRESS. It can translate a 10303-21 file into the equivalent terms of the Application Elements of the ARM Information Requirements. It also allows test data to be input using ARM terms and then translated into the corresponding 10303-21 syntax. The tool allows the ARM application element view to be used during semantic analysis of 10303-21 exchange files to verify that the input semantics have been encoded correctly. The data creation capability allows initial test data to be created for use in both interoperability and conformance test suites.

The Geometry Analyzer - This tool verifies the semantics of the geometry portion of a STEP file.

Figure 8-7 shows how each of these tools can be used to enhance the effectiveness of interoperability testing. The coverage analyzer examines the actual exchange files to determine the extent to which ISO 10303 has been covered during the interoperability testing. Conformance test suites contain test data designed to cover the entire scope of the standard; however, if the exchange files used in interoperability testing are only generated by IUT preprocessors, it is unlikely that all test purposes will be covered. The coverage analyzer can identify specifically what parts of the standard have not yet been exercised.

The STEP File Checker has proven to be an invaluable tool in interoperability testing. It performs both syntax and structural analysis on the 10303 exchange files to verify that they are correct. Finally the ARM/AIM Browser and the Geometry Analyzer can be used to perform semantic analysis on the 10303-21 exchange structures.

NIST also developed NIST Expresso [158], which is a language environment for ISO10303-11 that provides tools to aid in developing and validating EXPRESS information models and representative data sets. NIST Expresso is available as an executable that runs under Microsoft Windows 95 and NT operating systems. The downloadable PC executable allows the user to specify and incrementally modify an EXPRESS information model for analysis and validation. It may also be used to build representative data sets for the subject schema. To date, NIST Expresso is in beta-test status. Some testing has taken place against ISO 10303-202, -203, -213, and -214. NIST Expresso was also used in developing ISO 10303-302 Technical Report [159].

8.8 FORMAL CERTIFICATION OF PRODUCTS

US PRO has initiated a program to certify ISO 10303 AP implementations. The program, which will be brought forward for accreditation, is one of the first certification activities to be conducted on-line using the testing technologies and tools developed jointly by NIST and ITI. The testing program is comprised of a certification body, testing laboratories, and a certification control board. US PRO will act as the Certification Body and will issue certificates of compliance for those implementations to successfully complete ISO 10303 testing requirements (Figure 8-8). The initial testing service will test implementations for ISO 10303-203.

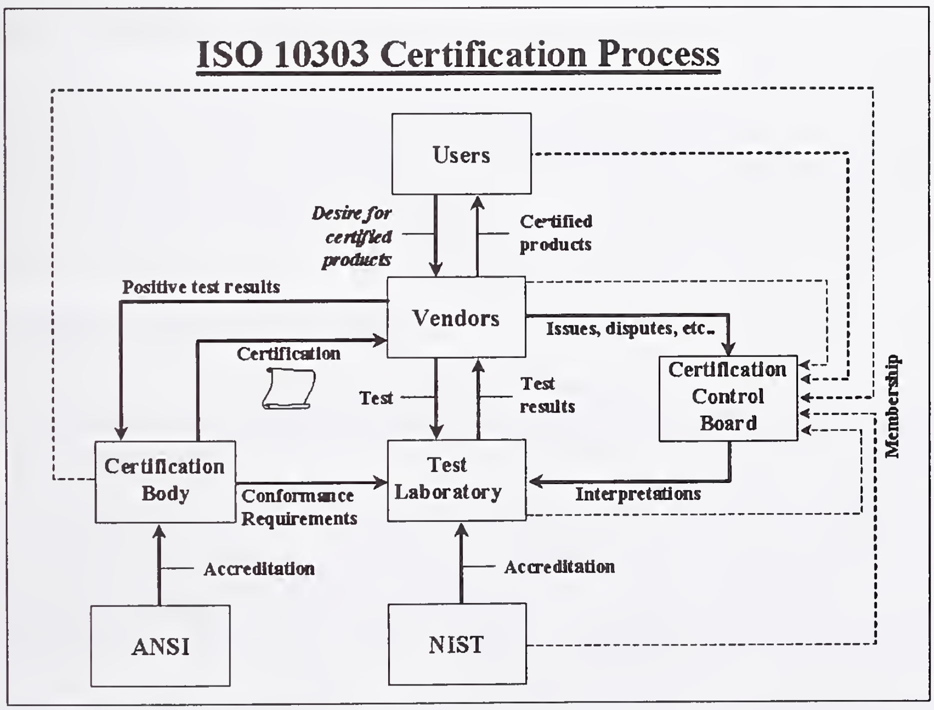


Figure 8-8: ISO 10303 Certification Process

The need for certification testing has its roots in the user community. It is intended to provide assurance that products comply with the requirements of ISO 10303. The availability of certified systems will encourage user companies to establish with their business partners product data exchange policies that are based on international standardized solutions, thus reducing dependency on expensive custom exchange or proprietary solutions [160].

For the last few years, PDES, Inc. has been running an activity known as "STEPnet." Started in 1995, and known then as "Plugfest," it is a group of CAD vendors and second- or third-party software developers conducting interoperability testing over the internet. The STEPnet goals are:

- Implement functionality for today's needs.
- Identify functionality for tomorrow's needs.
- Avoid roadblocks by establishing agreed upon approaches.
- Increase user confidence by providing system and AP interoperability testing.
- Implementing new functionality cannot adversely impact existing implementations [161].

ProSTEP, in Germany, also has a testing service for vendors. Primarily focusing on the implementation of ISO 10303-214, ProSTEP calls their testing activity "Test Rally." STEPnet and Test Rally have provided an informal means to test the benefits of conformance testing, while also proving the interoperability of the ISO 10303 implementations.

8.9 TESTING BENEFITS AND COSTS

Testing is critical to ensuring interoperable products, but it can add time and cost to the standards' and products' development processes. Successful interoperation is an essential benefit of standards-based products. Users expect that implementations, which claim support of standards, interoperate seamlessly with each other. Various forms of testing are employed routinely to assist the users in determining whether the products they purchase will interoperate. The methods and process of conformance testing affords several benefits to developing interoperable systems:

Early detection of errors – Conformance testing detects interoperability problems early in an implementation's lifecycle, when the cost of repair is significantly lower.

Optimized test cases – Conformance testing uses a carefully constructed set of tests designed to maximize coverage of the most significant inputs and states, while minimizing unnecessary test pattern and sequence redundancies.

Issue resolution – Conformance testing detects early implementation problems of developing APs, and encourages feedback of the necessary corrections early into the standards process.

Initial confidence – In the early part of a standard's lifecycle, conformance testing provides initial confidence and momentum for product development.

Although conformance testing is critical to assessing interoperability it takes time and increases the cost of developing standards and products. There are several ways that testing adds to the time and cost of development:

Developing the abstract test suites – Developing a complete test suite can be costly since it occurs in parallel with the developing AP and at a time when there are fewer tools available for generating valid data.

Developing testing tools – Testing requires the development of test tools for administering the tests, analyzing the results, and generating reports. Developing and maintaining such tools is a major resource commitment. It is usually an investment that is never financially recoverable.

Inputting data, running tests, and analyzing results – It takes time and resources to run the tests and analyze the results. This occurs during the STEP product development as well as during conformance testing. The frequency of testing is typically higher during the early phases of the implementation's lifecycle, though the tests tend to be less exhaustive.

Even prior to formalization as a service, NIST and ITI have already identified hard core dollar savings by vendors. Through conformance testing in the AutoSTEP and STEPnet efforts, significant savings were recognized by the participating CAD vendors. From twelve rounds of STEPnet testing, 10,000 unique structure violations were identified. Because conformance testing helped identify problems and cause early in the implementations' development, as much as $60 million savings were realized through intervention!

Traditionally, conformance testing is done near the end of the development phase of a product's lifecycle. At this point vendors are normally in a hurry to get their product into the market. They are also typically near the end (or over) of their development budget. The prospect of transporting their systems to a testing laboratory, and having their systems tested (only to find that they failed in one or more aspects of the standard) is truly frustrating for most. Further delays and costs are incurred as the vendor is forced to make changes to the implementation, which are much more expensive at this stage in its lifecycle. Vendors also trust their own testing facilities implicitly over those offered through an independent laboratory.

Ideally, the vendor should know before going to the testing laboratory whether its implementation would pass the conformance tests. This can only happen if the vendor has access to the test suites and tools so that conformance testing can be incorporated into their own product development process. One clear way to reduce the cost and increase the value of the investment in conformance test suites and tools is to make them more widely available on common platforms. This would make them more accessible to anyone who needs to develop an implementation. This not only reduces the unnecessary hurdle at the end of product development imposed by traditional conformance testing, but it also reduces the burden on the vendor in developing its own testing suites, tools, and laboratory facilities. By finding ways to apply these same tools beyond strict conformance testing, one can increase the overall benefit of these tools to the marketplace. Broad application of common testing environments and tools can improve STEP products and reduce development costs for those products. Such broad application also reduces trade barriers, thus allowing better international market support for a conforming implementation.

8.10 CONCLUSION

Conformance testing provides many advantages to developing standards-based products. With foresight, SC4 made conformance testing methodologies and requirements an integral part of the ISO 10303 standards' development process. STEP developers devoted two entire classes of parts to conformance testing (30 and 300 series of parts), and STEP APs explicitly state conformance requirements. The enabling technologies for conformance testing have been standardized for ISO 10303 and test procedures, and executable test suites are derived from these technologies. To aid this process, executable support tools have been developed by NIST and its partners.

The cost of conformance testing can be reduced through the development of tools to automate the manually intensive parts of the testing process. The costs of developing conformance test suites and tools can be further amortized by using them in other types of testing such as interoperability testing. Using tools and methods developed for the more formal conformance testing environment in an interoperability testing scenario results in a more robust testing method. Not only are the results of the combined approach better than either approach in isolation, but the use of tools greatly reduces the time and cost of performing these tests. The net result is that better implementations can be brought to the market in less time and for less cost. Informally, the AutoSTEP and STEPnet projects have already yielded great cost savings from applying conformance testing. NIST hopes that, by establishing a national certification program with bilateral agreements with other countries who have a vested interest in ISO 10303 implementations, worldwide recognition of ISO 10303-conforming implementations will be established.

What has been perhaps the most beneficial aspect of NIST's leadership in developing a conformance testing methodology and accreditation program is the exposure NIST has gained in software testing. NIST has had, from its start in 1901, a prominent role in the nation's measurement infrastructure as it applies to the seven basic international system of units (e.g., length, mass). With STEP conformance testing, NIST has seen a growing importance of its role in the ever-increasing demand for information technology metrology. NIST will continue to strive to extrapolate the results of the STEP testing methodology into the broader information technology arena.

CHAPTER 9

MANAGING THE PROCESS TO ACHIEVE THE PRODUCT -- STANDARDS

9.1 OUR MEANS TO AN END -- ORGANIZING AND OPERATING SC4

ISO TC 184/SC4 is a subcommittee under ISO Technical Committee 184 (Industrial automation systems and integration), and is responsible for industrial data standards other than those directly related to electrical or electronic standards. Current standardization efforts within the SC4 domain include STEP (ISO 10303), Parts library (ISO 13584), Manufacturing management data (ISO 15531), and Oil and gas (ISO 15926). As you read this chapter, it is important to realize SC4 is responsible for multiple standards under its domain, although this document focus has been on STEP. Chapter 9 covers the methods, work force, materials, and tools that contribute to the standardization process of STEP within the international standardization community.

One should approach this chapter with the understanding and appreciation that EVERYTHING surrounding the standardization of STEP is huge in magnitude and done with a respect for bigness and complexity. The quantity of meetings (at least three times a year for a full week), the number of technical experts at any given meeting (200-250), the unbounded scope of ISO 10303, and the complexity and size of any given 10303 part all contribute to the need to invent innovative ways to conduct business. SC4 requires more stringent quality requirements than ISO does for its standards. APs can be thousands of pages in size and require more complex standardization mechanisms and processes to ensure a good product than most of the SC4 standards. This chapter is not intended to reflect the naive belief that all aspects of the ISO TC 184/SC4 effort are unique. Where possible, ideas from other standards development communities were leveraged and adapted; however, in many respects, the way ISO TC 184/SC4 approaches an issue often seems the exception not the rule. Because other ISO standards are shorter in length and smaller in scope, and ISO subcommittees meet with less frequency and with a smaller number of technical experts, SC4 has had to find innovative ways to augment the traditional standardization process.

One of the problems innate to the ISO TC 184/SC4 community has been its inability for more than the short-term to define its scope and build an organization to support that scope. There is no single source to point to for this problem. It stems in part from the subcommittee's huge proportion and interpretation of scope at its birth and the belief that one standard (one work item) would meet that scope; and in part from the simple exponential growth of information technology. No one is able to predict completely what the future will bring one, three, or ten years from now. This means that one can anticipate further changes in the SC4 organization although it is not known what those changes will be. Figure 9-1 depicts a timeline showing the evolutionary organizational structure within SC4, as well as the subcommittee name changes that has occurred as the role and purpose of the subcommittee also evolved over time.

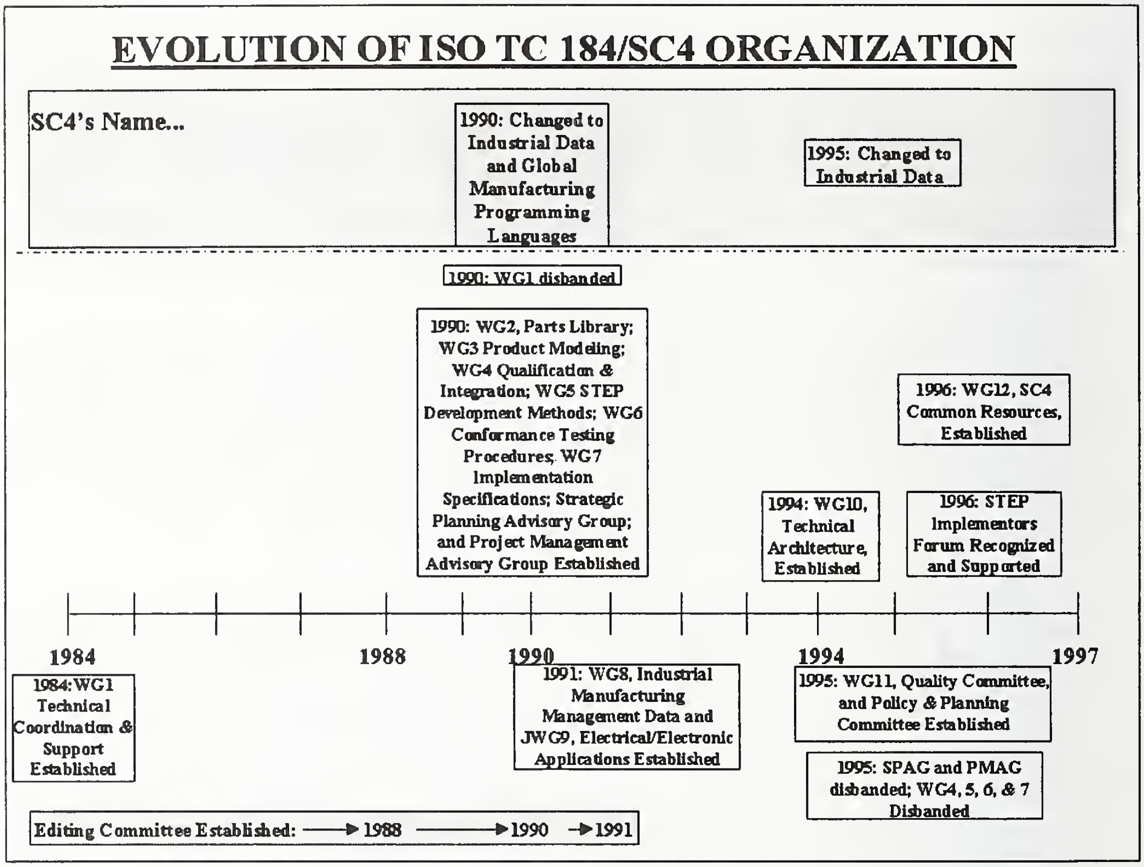


Figure 9-1: ISO TC 184/SC4 Evolution

Many ISO TC 184/SC4 resolutions affecting the organization structure offer an historical perspective of SC4's constant struggle for the "right" organization and the "right" title.[31] ISO Directives Part I [162] assumes an editing committee is part of the generic make-up of any subcommittee. This has never been assumed by SC4 as is noted by the recurring theme in SC4 resolutions:

- #30: create an editing committee (1988);
- #75: create an editing committee (1990); and ...
- #98: create an editing committee (1991).

However, it is apparent that SC4 P-members as a collective, resolution-making force recognize and respect the importance of building quality documents (i.e., resolution 130 in 1992).

Another interesting characteristic and recurring organizational theme within SC4 is the tendency to create advisory groups. Membership in early advisory groups, such as the Project Management Advisory Group (PMAG), was open and members discussed the overall work of SC4. Currently, a handful of senior advisors is selected to support the Chair and Secretariat. As the SC4 national membership grew, and as the breadth, depth, and volume of projects grew, SC4 recognized implicitly that some advisory capacity must be added to assist the SC4 Chair and Secretariat. During the period of 1984-1995, one individual at NIST served as both the SC4 Chair and Secretary, and two part-time administrative staff comprised the remainder of the Secretariat support staff. To support this staff, the SC4 advisory bodies served a dual-role: providing strategic, long-term focus for the subcommittee, and assisting in the daily project management and technical issue resolution. As SC4's project load continued to grow, NIST recognized a need to beef up the Secretariat support. Today, NIST co-sponsors the SC4 Chair from U.S. industry, and provides a part-time Secretary and five part-time administrative and technical support staff. The role of an

[31] These complete resolutions can be found in Appendix C under 'Chapter 9'.

advisory group to the Chair and Secretariat now has a different meaning: assist in scoping out the long-term strategic perspective and help sort out any political issues. This assistance is carried out while the Chair and Secretariat conduct the daily project management and technical issue resolution.

Howard Bloom recalls... The U.S. TAG to SC4 took the organization very seriously. The issue was – how many working groups for which we should we try to seek the chair? Long debates were held before we made our final decisions. One humorous note was the selection of the U.S. Policy Management Advisory Group (PMAG) representative to SC4. At the time, the IPO Chair position was held by Steve Ray who had agreed to a temporary nine-month appointment. In coming up with names for the ballot, Dr. Ray was nominated by name; the generic position "IPO Chair" was also nominated. It is not often that one gets to run against oneself! As it turned out, the IPO Chair position won and the IPO Chair continued to occupy that position until the PMAG disbanded several years later.

By 1998, SC4 has seven working groups and a quality committee to manage the more than 100 ongoing SC4 projects. Through all its potential stumbling to create the perfect international standards-making machine, the SC4 community has been earnest in effort and devoted to developing what it believes to be good results. It has attempted to better itself and its standards products by placing more stringent requirements on itself than those required by ISO. The subcommittee has invented ways to process the standards effectively and with quality output, while at the same time continuing to tap the tacit knowledge of its wise technical experts. This section highlights a few of the ways to do standardization work which were invented within SC4 or borrowed from concepts invented by others.

9.1.1 STEP Standardization --- Process Features

A well-defined organizational structure and adequate procedures for the flow of the work are the minimum requirement. The highly dependent aspects of the work demand free and open communication between all the individuals involved. ... Very early in the work of SC4, it was recognized that the volume and the complexity of the SC4 standards to be developed required supporting tools, and EXPRESS was developed to enable a formal description of the information content of the standards. Structures, rules, policies, and style guides are required in order to obtain a maximum of conciseness, uniformity, and clarity; and to avoid duplication of nearly identical definitions [163]."

Brad Smith recalls... The size of the Tokyo draft amazed all who saw it - over 1000 pages - and SC4 and NIST began to seriously worry how to record, sort, distribute, analyze and respond to each of the ballot comments that were expected. So SC4 committed to using a database and made up an executable disk which was sent to each P-member country for the ballot. With the conclusion of the ballot there were 2300 comments. A huge weekend working session was held in Frankfurt, just before the SC4 meeting was to start that next Monday. The job was to complete the task of entering all ballot comments into the database. Each delegate had brought a computer with him, and they were arranged conveniently around the room. I remember using whatever computer was free at the time, but I would occasionally get an error message and have to stop to get help. Even though all computers were running the same database software, the error messages would come out in German, French, Dutch, etc., depending on whose computer I was using!

Easily stated, but much more difficult to execute. SC4 and its supporting Secretariat (NIST) have implemented several means to improve work flow and open communications. It was a long battle starting about 1985 to take proven methods from the research community and transition them into the traditionally paper-based standards community. Many anecdotes have already recounted incidents along these lines, particularly with respect to ANSI and ISO.

The Secretariat was the primary force advocating change from the traditional paper-based environment that existed at the first meeting of SC4 in 1984 to the present electronic environment. NIST should take the credit for this revolution. SC4 pioneered many of the "firsts" in the standards community, things that have been adopted by others and are accepted as commonplace. SC4 led the ISO community and was most likely the first standards committee to

- Have a majority of its participating experts using e-mail.
- Use e-mail exploders for routine electronic communication.
- Have its own approved web site.
- Develop automated tools that help build standards and check for conformance.
- Establish an electronic repository for all working documentation (SOLIS).
- Use the web at a standards' meeting to make available minutes from the previous day.
- Publish a computer-related standard in a computer-sensible form.
- Convince ISO that a diskette was essential for publishing normative information along with the hardcopy of the standard.

Some of these accomplishments seem odd in light of the many computer science and language standards committees (e.g., ISO/IEC JTC1) that had been in operation for many years before SC4 started. Yet in instance after instance, the SC4 Secretariat could not find the tools it needed to solve its communication and information dissemination problems. NIST developed EXPRESS checking tools (See section 9.2 below) because the developing SC4 standards are complex and also to ensure the SC4-prescribed quality of EXPRESS was produced. These tools are applied by the standards developers during national technical reviews of the draft standards, and by the NIST SC4 Secretariat to ensure the syntactical integrity of a standard before it begins its balloting cycle.
Use of WWW, E-mail, Exploders

"Our committee [SC4] has tried to develop good project communications as an essential factor in building close teamwork among our distributed group of experts. We continue to explore new forms of electronic communication designed to foster a more efficient working environment. Some of these include:

- E-mail links among all those who can get direct access via Internet, Bitnet, etc. and dialup serial access for a number of others via commercial E-mail accounts, for instance MCImail;
- Alias mailing lists to reach selected groups of people;
- An archive of all significant project documentation; [and]
- An E-mail archive-server to fill requests for documents [164]."

It seemed an appropriate and interesting exercise in electronic communication growth, to compare some of the 1992 statistics offered by Mr. Smith [165] against today's SC4 use of digital exchange. Table 9-1 shows the comparative categories offered by Mr. Smith in 1992 against similar information available in 1997.

Category	1992 Total	1992 E-mail Capability	% of 1992 Use	1997 Total	1997 E-mail Capability	% of 1997 Use
Working Group Conveners	8	6	92	8	8	100
Working Group Deputies	2	1	50	4	4	100
Advisory Group Chairmen	3	2	66	Not Applicable	Not Applicable	
Project and Team Leaders	42	41	97	66	66	100
Part Editors	29	24	83	48	48	100
Chairman and Secretariat	3	3	100	8	8	100
P-member Bodies	Not Available	Not Available		19	15	79

Table 9-1: SC4's use of digital exchange

In only five years time, a technology shift has occurred and many more people are now able to access the Internet and leverage its services through electronic mail and the World Wide Web. SC4 currently has over fifty special interest group (SIG) exploders established to support daily interaction worldwide. Many of the SIGs use the exploder to broach questions, suggest strategies, and work out technical issues. Having this cheaper, faster, more efficient means for communication has helped reduce travel expenses and standards development time. The increased use of electronic communication enabled by the Secretariat has allowed the committee to reduce the number of face-to-face meetings even though the committee's workload is increasing. NIST provides the technical infrastructure and administrative support to help develop and implement these critical means for faster communication. Meetings have already been reduced 25% a year, from four meetings to three, thus saving participating companies and standards bodies substantial travel costs.

Joan Wellington recalls…Today it seems commonplace but during the early 1990s the technology was used very well "before its time." During the publication of the Initial Release, entire standards documents were sent via e-mail and of course, there were the technical discussion groups that saved thousands of dollars in travel money and let members participate in discussions on their own schedules. Correspondence on a one-to-one basis via e-mail was also very important – the difference in time among the various participants was never a big problem. I believe the use of e-mail played a very large role in the creation of STEP.

NIST also sponsors the electronic document and configuration management site for product standards and supporting development methods. This service is known as the SC4 On-Line Information Service (SOLIS). Essential to the productive effort of worldwide standards-development, SOLIS offers a repository of information for the SC4 community. SOLIS started in 1990 with funding from the CALS Program [166].

The information made available through SOLIS is entirely public by permission of ISO. Only pre-Draft International Standards (DISs) documents and supporting software tools, meeting announcements and minutes, working group documentation, project information, and other documents such as development guidelines, issue logs, and STEP-specific style sheets, can be obtained through SOLIS. In addition, ISO has granted permission to allow the AP EXPRESS specifications and Units of Functionality to be available on SOLIS, even those from published international standards. Other copyrighted material is stored in a separate password-protected area and is not available to the public. It *is* available to the SC4 community for furthering other standards' development that is based on initial releases. The impact of SOLIS on the SC4 community's work is described later in this chapter.

9.1.1.1 Infrastructure Functions

As early as 1992, SC4 recognized the merit of quality assessment of its complex standard parts. SC4 gave particular attention to the STEP application protocols. The purpose was to ensure such complex documents could best meet the STEP design goals of compatibility with other standards, minimize redundancy, and maximize completeness for exchange and archiving. The quality approach was two-fold:

- Define and document the methods, metrics, and procedures.
- Educate the developers so that they could take their learned skills back to their project and apply it to their particular part(s).

In the fall of 1995, ISO Central Secretariat staff educated the SC4 Chair and Secretary about the use of standing documents within a subcommittee. Subcommittees may develop and approve such documents to facilitate standards development within the subcommittee. This approach seemed to provide a tidy way of adopting and deploying the historical efforts of a working group to the subcommittee. Since 1995, seven methods documents have been adopted as SC4 standing documents. NIST played a contributing or leading role in all seven documents.

The procedural aspect in SC4 to educate the developers in any procedural changes was not well documented prior to the use of standing documents. In the case of AP development, the procedures are highly resource-intensive. To manage integration, interpretation, and general quality of a given AP, the requirement imposed on the AP Team was to provide resource(s) to participate in any of several SC4 qualifying processes. The commitment for each team was approximately 900 hours; however, it was unclear as to when one was expected to provide those hours, or against which tasks one should apply the hours. For those AP developers who provided their appropriate resources, the quality of the standard as a product was notably better than for those who chose not to provide resources.

Several lessons learned have accumulated from these quality methods and processes over the years:

- It is difficult to get and retain resources to work on defining quality methods, metrics, and processes; most interest lies in developing the implementable portion of STEP --- the application protocol. (This theme has been highlighted several times in the preceding chapters.)
- If one wants a requirement to be understood, one must clearly document and inform the audience from the start.
- If one has a constraining procedural or methodological requirement without a means to enforce it, few will actively participate to meet the requirements.
- One should not create requirements that cannot be documented for consistent and repetitive application, e.g., integration and interpretation discussed in Chapter 4.
- When one's resources are severely constrained (such as those within Quality Committee) on a critical path for standardization completion, one creates a severe bottleneck in the process.[32]

NIST has tried to apply these learned lessons as it has carried out its SC4 infrastructure support functions. Besides authoring most of the methods documents, NIST provides the leadership of the Quality Committee and resources to the qualification process. As the workload continues to expand with the increased volume of work within SC4 the Secretariat, even with other NIST resources, is unable to alleviate the bottleneck on its own. The Secretariat continues to work with the SC4 community to draw on other national resources.

9.1.1.2 STEP Implementors' Forum

Although the STEP Implementors' Forum is only an informal part of SC4, it is important to recognize its function within the organizational structure of SC4. The IPO led the effort to create this Forum as an opportunity for implementors who build ISO 10303 processors and tools to meet and discuss issues surrounding those implementations. Such a Forum helps a standardization group keep its end goals in perspective:

- implementing the standard
- what is the impact to implementors when the standard is revised

The efforts within the Forum are well coordinated with SC4 liaison organizations that work directly with the implementors. The Chair and Secretary have found it useful on several occasions to use the Forum as a way to get the pulse on the STEP implementation world. In response to a call for position papers, NIST drafted the Forum's initial proposal as the United States submission (SC4 Resolution 216) [167]. Many of the concepts from NIST's contribution were used when establishing the Forum.
Change Management of the Standards

Once years of labor bear fruit by producing an international standard, the standard's project team often disbands, having "done its job." Having no project team has caused some difficulty within the SC4 community in the processing and handling of omissions, errors, and ambiguities within a particular standard. Despite the rigorous quality regulations established within SC4, such discrepancies happen among most international standards. Until someone not involved in developing the standard begins to interpret its content, or apply the standard in a real life industrial setting, no one can anticipate all the necessary fixes required.

Modification to an international standard comes in three forms: a technical corrigendum, an amendment, or a revision. A technical corrigendum is issued to correct a technical error or ambiguity that could lead to incorrect or unsafe application of the standard. An amendment alters or adds to previously agreed technical provisions. A revision results in the publication of a new edition of the standard. A revision is developed when the extent and scope of the changes make it impractical to publish changes in the form of an amendment or technical corrigendum.

As part of the Quality Committee, SC4 established a Change Management Team responsible for reviewing preliminary work items and new work items [168] that propose changes to existing international standards before these changes are submitted to SC4 for approval. To support change management, SC4 established a Standard

[32] Current SC4 qualification practice requires each draft standard to undergo a review against prescribed Quality Committee procedures.

Enhancement and Discrepancy System (SEDS) in 1995. The SEDS process manages corrections and additions, and tracks the progress of these modifications until they become part of the appropriate international standard through the ISO publication process. NIST, in collaboration with the IPO, played a leadership role in proposing the initial SEDS procedure for the SC4 community's adoption in 1995.

This procedure was based on earlier work done within ISO TC 184/SC5[33]. The SEDS's initial design and intent was to quickly process, for the benefit of the implementors, the discrepancies as they were discovered. Unfortunately, the many faces of reality have forced the SEDS process to continue to evolve and be re-examined since its inception. The most recent procedural changes seem more likely to yield success in the total SEDS process. Timing has historically been, and continues to be, an issue. Does one process a SEDS report individually to meet the solution in a short-turnaround approach; or hold the individual SEDS report for some unknown quantity of time to see if other related reports, or reports against the same part, may also be submitted?

Once a SEDS report is closed by SC4, implementing the change in the report becomes an issue. Until the correction or enhancement has been standardized officially through the ISO consensus process, who owns the liability if an implementor should use the SEDS solution --- the Secretariat for posting the solution on SOLIS? The user for requiring the implementor to change? Alternatively, the implementor for making the change in the first place? The SC4 Quality Committee Change Management Team and the Secretariat continue to revisit the SEDS procedure to respond and resolve the known issues, and to make the overall SEDS process more efficient and effective.

9.1.2 From Tacit to Tangible Transfer

"SC4 shall follow the principles of organization in industry, where, for motivational reasons, functions such as quality control are delegated to the operational units. This demands adequate training, provided centrally for consistency [169]."

One of the most impressive phenomenons of ISO TC 184/SC4, if judged against normal ISO subcommittee standards, is its continued high meeting participation volume. If one has ever attended a week-long SC4 meeting, several things are of immediate note:

- The number of people with an interest to develop product data standards (approximately 200-250).
- The quantity and diversity of meetings on any given day (20-30).
- The length of any given meeting day --- starting with 0700 breakfast meetings and often going well past the 12-hour work day.

Since its inception as a subcommittee in 1984, ISO TC 184/SC4 has remained active and its membership levels relatively constant. It is not unusual to find attendees who have been participants for close to a decade, and one can note with silent appreciation and respect as the "gray hairs" of maturity and wisdom become more pronounced over the years of involvement. The level of effort during any of the week-long sessions is awesome, and the enthusiasm among the working teams is quite contagious. Individual experts devote tremendous amounts of personal time to the common cause. Some have even taken personal vacation time to attend the meetings when their companies lacked a travel budget; others literally changed companies in order to continue with the SC4 effort.

To show a comparison of breadth and depth of participation, Table 9-2 shows the attendance at an SC4 meeting in London in 1992 [170] as compared to more recent SC4 meeting attendance during the same time of year – the "Spring" SC4 meeting[34]. The first nineteen countries represent the participating member countries during this time.

---

[33] ISO TC 184's Subcommittee SC5 focuses on Architecture, communications and integration frameworks. Collaborative work between SC4 and SC5 is discussed at more length in Chapter 10.

[34] Current practice is to hold a meeting in the January-March, May-June, and September-October timeframes. There is sometimes slight variation to this routine.

Attending Country	London, UK 1992-07	Kobe, Japan 1996-06	San Diego, USA** 1997-06
Australia		1	2
Belgium	1		1
Brazil			
Canada	1	1	1
China		1	1
France	20	9	14
Germany	29	23	24
Hungary			
Italy	3	2	1
Japan	4	92	40
Korea, Republic of		5	1
Netherlands	3	2	6
Norway	6	3	7
Romania			
Russia			
Sweden	5	4	4
Switzerland	3	2	3
United Kingdom	30	19	24
United States	54	41	134
Denmark			1
Finland		2	3
Spain			1
TOTAL	159	207	268

** Joint ISO SC4/IPO Meeting

Table 9-2: Examples of SC4 Meeting Attendance

Many of the same dedicated players participate meeting after meeting, year after year. Nevertheless, it is critical to tap into the tacit knowledge of the hard core, long term, experienced experts and transfer this information to the new participants. The knowledge and lessons learned must outlast the individual who may have experienced it first hand. To encourage such information transfer, several initiatives were started and maintained across the years. Several of these initiatives were U.S.-led.

Since the United States continues to have a high investment in the success of product data exchange standardization, the ANSI Administrator, US PRO, routinely offers to host the SC4 meetings (historically at least once annually). When it serves as host, these SC4 meetings are co-hosted with the parallel U.S. ANSI organization, the IPO. There are several information deployment techniques that have been established by the IPO over the years, to the benefit of both the SC4 community at large and the U.S. participants. The "bait, educate, and capture" technique started on the weekend preceding the beginning of a joint meeting. This technique has helped remove some of the mystery for a newcomer, lured into the SC4 or IPO flock, and allowed smoother operations during the week of meetings. The approach is somewhat adaptable to a location, but usually takes the form of a topical one-day workshop or several technical training sessions covering topics such as "what is STEP." More recently, other country hosts have also sponsored pre-meeting workshops to attract new regional interest on the topic of product data exchange standardization. The preceding weekend of a joint ISO/IPO meeting always wraps up Sunday evening with the Newcomers' Orientation. Through its IPO participation, NIST also started a complimentary "mentoring" support program to further alleviate confusion for the new participants. At midweek lunch, the newcomers were encouraged to sit at reserved tables with the IPO and SC4 leadership to share their experiences thus far.

As the week begins to unfold into 20-30 simultaneous committee meetings daily, additional technical or informative tutorials are also offered. These are routinely evening events and offered by the dedicated volunteers within SC4. These sessions assist in continuous process improvement, better awareness of new or developing projects, and common understanding and appreciation of related standards development efforts.

Other means employed to educate and inform SC4 members and technical experts are:

- News releases: information blasts sent to the largest SIG exploder about current project status, upcoming meeting agendas, or other information of importance to progress the standards.
- SOLIS: Access to the EXPRESS portions of the developing and mature parts.
- Boilerplate language, templates, style files, and standing documents: tools to ease some of the development pain of the project leaders and part editors.

The size of the SC4 organization and the number and complexity of the SC4 standards have resulted in the need to define supporting processes, structures, rules, policies, and style guides. The intent of such guides is to obtain conciseness, uniformity, and clarity while meeting industrial application requirements. The phased development approach used by SC4 to publish its standards in a series of parts, while still maintaining an integrated whole, also contributes to this requirement. Achieving a quality result necessitates that these processes, structures, rules, policies, and style guides be documented, understood, and accepted by all of the SC4 standards developers.

In some cases, when it is expected that the methods developed by SC4 have general use outside of the SC4 community, this documentation will take the form of ISO standards or ISO Technical Reports. In cases where SC4 determines the applicability to be limited to the SC4 community, these documents will take the form of SC4 Standing Documents. As a means to document and actively transfer tacit knowledge across the SC4 community, several SC4 standing documents have been created. These, as mentioned earlier, contribute to the technical infrastructure and fiber of SC4's quality procedures.

9.2 EXPLOITATION OF INFORMATION TECHNOLOGY

The success of STEP relies heavily upon the knowledge and experience of its technical experts. These experts fulfill the essential role of ensuring the technical accuracy, completeness, and applicability of STEP to the industrial communities it serves. Nevertheless, the size and scope of the standard make it equally essential to seek opportunities to exploit information technology (IT) to streamline and remove barriers and impediments that impede standardization. This section addresses the ways in which STEP is doing just that. Three areas of innovative IT application are discussed: AP development, use of the Internet, and support for SC4 administration.

9.2.1 AP Development Environment: Maximizing Productivity and Quality

To aid in the STEP AP development process, NIST developed components of an integrated software environment known as the Application Protocol Development Environment (APDE). The goals of the APDE are to

- Increase the productivity of AP developers.
- Improve the quality of the resulting draft standards.
- Provide software tools and services customized to the needs of STEP AP developers.

The APDE is intended to support the time-critical, resource-intensive tasks in AP development. Such development can be improved through information technology services and computer automation. To this end, the APDE architecture includes three main areas of functionality: document preparation and publishing, repository services, and EXPRESS services. Figure 9-2 illustrates this architecture.

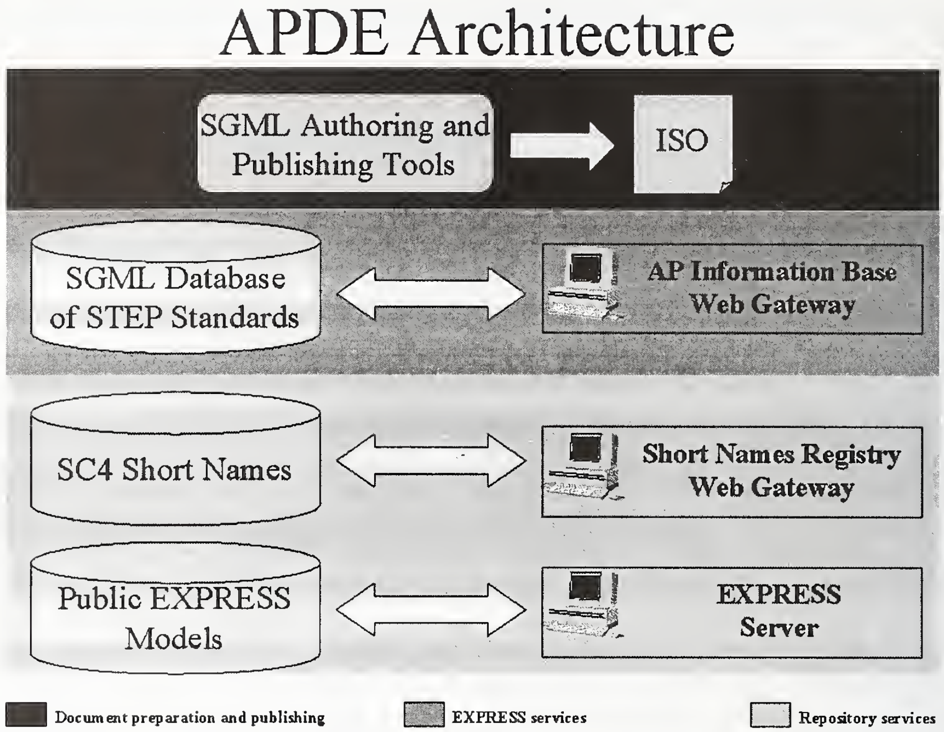


Figure 9-2: APDE Architecture

9.2.1.1 The Birth of the APDE - Addressing the Mission Critical Tasks

Initial APDE efforts focused on enabling the mission-critical AP development tasks. These were identified as the STEP interpretation (see Chapter 4) and validation processes. These tools are still in widespread use today, and provide the ability to verify the syntactical correctness and validity of information models using EXPRESS. Developments in this area include an EXPRESS parser (fedex), short-to-long (shtolo) form generator, the STEP Class Library (see Chapter 5), and the Data Probe, as described in Table 9-3 below.

Tool name	Description
Data Probe [171]	Editor for data described using EXPRESS
EXPRESS Pretty Printing [172]	Toolkit for formatting and printing EXPRESS objects
EXPRESS server [173]	Interface for remote access to EXPRESS-based tools
EXPRESS toolkit [174]	Toolkit for building EXPRESS-related software
Fed-X [175]	EXPRES parser
Short Names Registry [176]	Database of short names for each of the entity data types within each of the EXPRESS schemas
Shtolo [177]	Generates STEP EXPRESS annotated long form from short form
STEP Class Library [178]	EXPRESS-to-C++ translator
Transformr [179]	Data migration tool for evolving schemas

Table 9-3: EXPRESS-based APDE tools

9.2.1.2 Managing the Documentation Process

Later APDE efforts focussed on document management, specifically in the areas of document authoring, publishing, and electronic delivery. The objective of this document management effort is to identify and support target areas for process improvement and task automation to allow STEP experts to focus their efforts on the technical content of documents rather than on the documentation process itself.

Opportunities for process improvement in this area are many. The current documentation process has been cited as one of the most error-prone and time-consuming tasks in AP development [180]. This is largely due to the fact that STEP part editors use various proprietary systems that are not integrated or customized for use in STEP. Furthermore, the specificity and rigidity of STEP documentation requirements make it difficult for technical experts and quality reviewers to adhere to and enforce these requirements. Therefore, excessive man-hours are spent by both editors and qualifiers on checking document formatting and structure. In addition, the use of proprietary authoring systems for documents makes it difficult to find and retrieve information in STEP documents for reuse.

To address these problems, the APDE provided mechanisms to enforce structural documentation requirements automatically, generate the proper formatting, and verify those structures and formats. In addition, the APDE provided a World Wide Web-based system called the Application Protocol Information Base that makes STEP documents available in more useful forms and which increases accessibility to existing documentation. (See Section 9.2.2.3).

9.2.1.3 An SGML Environment for STEP

These new document management capabilities are enabled using an ISO standard called the Standard Generalized Markup Language (SGML) [181]. SGML is an ASCII-based markup language for describing the structure and content of documents in a computer-interpretable format. SGML Document Type Definitions (DTDs) are used to describe the structure and content of a given class of documents. Documents are tagged in SGML based on a DTD. This ensures content, structural accuracy, and consistency across documents, and allows context-sensitive, markup-based search and retrieval of the information contained therein.

NIST built an SGML environment for STEP that included all of the above described document management capabilities [182]. At the core of this environment are the NIST-developed DTDs for STEP documents that define the content and structural requirements of STEP documents. The use of SGML and DTDs for STEP enables complex documentation requirements to be enforced more reliably and efficiently. DTDs also enable sharing of documents across heterogeneous computer systems, and "intelligent" access to the DTD-defined structural components of STEP documents.

The NIST APDE project has concluded. Its legacy components include the STEP DTDs, the AP Information Base, an SGML publishing application, and several utilities for converting documents in proprietary formats to and from SGML. AP development teams have begun to utilize components of the SGML environment, and are seeing marked improvements in development time and document quality.

9.2.2 Providing International Access to Standards

In a large worldwide standards development effort accessibility to component specifications is an essential issue. With most participating countries having access to e-mail (Table 9-1), and a growing number having access to the World Wide Web, NIST is using both means to broaden accessibility to STEP and STEP-related information, tools, and services. Three main areas of use are the NIST EXPRESS server, SOLIS, and the APIB.

9.2.2.1 EXPRESS Server

NIST developed and maintains an EXPRESS server that provides e-mail access to EXPRESS-based tools and services for ISO 10303 development. Users simply submit an e-mail request that identifies the tool or service that is desired plus any required input information. The EXPRESS server then generates output and e-mails the user the results.

The EXPRESS server can be used to execute Fedex and shtolo, and to build a schema-specific Data Probe (see Table 9-3). Before NIST developed the EXPRESS server, users could only run these applications locally. Because these tools are Unix-based applications, STEP developers running PCs and MacIntosh equipment could not use the tools. The EXPRESS server alleviated this platform problem by providing remote access to the tools through e-mail.

9.2.2.2 SOLIS

SOLIS, introduced earlier, has had a major impact on developing the standard. It provides the means for geographically dispersed contributors to communicate, disseminate, and exchange pertinent information as the standard is developed. SOLIS "enhances the ability to gain consensus on (STEP) by expanding the availability of STEP draft standards, supporting documents, and software used by the community of experts who are contributing to the standard [183]." Today, SOLIS contains more than 9,000 files with an average of 4,000 unique users accessing its information on a monthly basis. The e-mail archive-server allows automated request and delivery services for those documents contained on SOLIS. SOLIS has provided the single most accessible and content-robust resource of SC4 information to date. Historically, the documents on SOLIS were accessed via ftp and electronic mail; however, since World Wide Web access to SOLIS has been implemented, access is even easier and data is available to more users.
AP Information Base

The Application Protocol Information Base (APIB) is a central repository, also located at NIST, which provides access to 10303 documents. Most of its current content is tagged in SGML to enable access to individual components of the documents. This feature distinguishes the APIB from SOLIS -- the ability to execute queries against the standard parts and get at individual components. This ability to access and reuse specific information is believed to be critical to the efficient and timely development of APs. Previously, users had no means of searching against the documents that comprise 10303 for specific information. This created a huge bottleneck in the development process. Developers were required to look for existing information manually in resource parts by literally flipping pages and often re-entering information if the format of the original document was not compatible with the document being authored.

The APIB provides the ability to execute a search against the entire repository of 10303 documents for a specific term, and because SGML is an ASCII-based standard, the information can be reused as-is. The use of SGML also enables users to narrow their search at various levels across and within documents. For example, a user can choose to search for a term across all 10303 parts, within a particular 10303 part or within a particular clause within a 10303 part. A user can also look for specific types of information, such as a Unit of Functionality or an EXPRESS construct. Users wishing to reuse the retrieved information can save query results to a file in its native SGML format.

Because there are no freeware SGML browsers yet available for the World Wide Web, query results are currently converted into HTML, which can be viewed with any World Wide Web browser. If native SGML browsers for the World Wide Web become available, the strategy will be to utilize these browsers to enable access to all DTD-defined components of the document directly from the SGML.

9.2.3 Supporting Administrative Functions

A third area in which information technology has been applied to SC4 is toward the performance of administrative functions. The SC4 Secretariat utilizes office automation tools to support its administrative activities that otherwise would be excessively time-consuming and prohibitively difficult to manage. Presently, the project management database is the primary utility.

With more than 100 active projects for ISO 10303 alone, NIST must continuously manage resources, schedules, leadership, progress, ballots, and associated ballot results for each and every one of these projects -- and through the usual four ballot cycles. By the nature of a voluntary organization, approximately 10-15% of the leadership changes every year. This requires a good record and maintenance for an historical perspective. NIST developed the SC4 project management database to support several of the Secretariat functions. It serves as the "life line" to project management. Key information can be requested and readily obtained by queries to the database. Relationships across parts can be easily established and schedules assessed to identify critical path requirements. Although having this database of information available for a multiplicity of query and task resolutions should be notable as successful management --- the Secretariat can only be as effective as the information is accurate. Since such capabilities for closer project management are relatively new to the SC4 culture, it has been a continuous struggle to disseminate to the many project leaders what is required of them, *AND!* have them successfully produce the information in a timely, repetitive, and consistent manner. Despite this learning curve within the SC4 community, the NIST Secretariat has already experienced timesaving in production overhead for many routine tasks because of this database management approach.

9.2.4 Exploitation Lessons Learned

In applying Information Technology to the standards development process, some results were not anticipated; however, lessons were learned from the experiences. These lessons are described briefly below.

- Simply developing or applying a new technology does not guarantee its acceptance into the user community. It is equally important to provide training to standards developers as they receive new and unfamiliar applications.
- In the rush to get tools out to customers, there can be a tendency to deliver tools before they are tested adequately. It is important to make sure tools work before distributing them to the general public. Establishing a beta test group that uses and provides feedback on work-in-progress applications can do this.
- A noble goal of the APDE was to provide every tool otherwise not provided by the commercial market to help AP developers be more productive. It is important to note that once the goal is accomplished, a high resource demand for support and maintenance of these tools continues to exist over time. NIST had to face reality about promises, both with regard to potential opportunities for commercialization and about its ability to maintain and support tools that are not likely to be commercialized.
- In an effort to leverage commercial products for STEP AP development purposes, it has become apparent that vendors have limited interest in supporting pre-standard specifications. Therefore, NIST needed to be prepared to develop, support, and maintain large portions of the APDE in-house. With limited funding and many priorities it was not feasible to continue the APDE as originally planned.
- The tools are only good if people are able to use them. Skill or technology may limit users, and the tools must support the least common denominator of both the skill set and the technology available to users.
- To increase the likelihood of commercialization of APDE capabilities, it is important to provide solutions with broad utility versus domain-specific application wherever possible.

9.2.5 Remaining Issues for IT Exploitation

Unfortunately, providing tools to support a developing standard is no panacea. Solution providers grapple with difficult issues for which there are no clear solutions. The major issues plaguing IT application to standards development are summarized below.

- **Copyright protection versus broad exposure**. There is a need to balance gaining exposure to the standard with protecting the ISO copyright. This issue is most problematic when using the WWW or SOLIS where special password protection capabilities have to be introduced to protect copyrighted information. Simultaneously, for outsiders to become familiar with the standard, at least selected parts of the standard must be made easily accessible; the number of potential, legitimate users should not be limited or thwarted by password-protected interfaces. The challenge is to balance public accessibility with ISO's right to protect.

- **Directives and methods keep changing**. Developing quality tools that support and enforce the specified directives is more difficult when the requirements are unstable. Unfortunately, tool developers have to shoot at several moving targets. This delays the delivery of tools to the end users and sometimes results in a mismatch between required and actual tool functionality. In addition, more time and money have to be spent on tool maintenance to support changing requirements. Therefore, it is important but harder to coordinate tool release schedules with the existing multitude and versions of directives and methods documents.

- **Migration to SGML**. The use of SGML promises great rewards for both its platform neutrality and its computer-interpretable format. However, both an inherent resistance to change among part editors and the unfamiliarity of the SGML authoring interfaces makes new SGML authoring environments difficult to deploy. Also, the complexity of the standard and the documentation requirements means that the structures represented in SGML are also complex, which further delays the impetus to SGML use.[35]

- **Lowest common denominator of capability inhibits information transfer**. As mentioned above, tools need to support the least common denominator of the technologies available to users. This restricts use of some advanced technologies such as asynchronous transfer mode or Java applets. These limitations can also constrain the performance and capabilities of the tools provided.

- **Handling and maintaining data in multiple formats**. Current legacy data is in several proprietary and defacto formats including: WordPerfect, Latex, and Microsoft Word. This presents a problem in several areas. First, the Secretariat must support all of these formats in multiple versions. As users' software platforms change, so must the Secretariat's. The Secretariat is faced with the difficult choice of deciding which software formats to allow. With each project having its own software preference, choosing a single format is nearly impossible. Secondly, because the data is stored in multiple formats, conversion to SGML is difficult. Multiple conversion strategies have to be employed to support all of the different formats. (This slowed the rate at which STEP parts were made available through the APIB.) This issue is further compounded when standards developers take five-ten years to develop an ISO 10303 standard part instead of meeting the current ISO requirement of three years.

- **Limited interest among commercial vendors to build tools supporting development of ISO 10303 or SGML applications**. Because both the STEP development and SGML user communities are relatively small, vendors have little incentive to build tools to facilitate the development of these standards. There are an increasing number of applications that support SGML, but still few. Fewer still are applications available that support STEP development. Therefore, STEP AP developers can leverage only a small pool of commercial or public domain tools.

9.3 LEVERAGING HUMAN RESOURCES

While IT plays an important role in successfully developing the standard, it is also critical to ensure effective utilization of human resources. Both internal and external participants to standards development play a key role in ensuring the overall quality of the standard by providing uniquely human contributions to the standard -- contributions such as knowledge, expertise, vision, or even political clout. This section focuses on the external contributors to STEP. The following human resources are considered: tool developers and support staff, industry liaisons, and collaborative SC4 partners.

9.3.1 Providing Technical Support for Tools

Merely releasing tools into the end user community is not enough to ensure they are applied effectively to develop standards. This is especially true given the new approaches being employed, such as the use of SGML. A major part of IT services is the technical support provided by people. NIST offers direct technical support to AP developers to help ensure the proper and most effective use of the tools. E-mail exploders at NIST have been set up to handle incoming questions, bug reports, and requests regarding tools. NIST APDE tool developers have also attended most IPO and U.S.-hosted SC4 meetings to provide demonstrations and training on new applications. In

[35] Chapter 10 identifies additional applications of SGML in a STEP environment.

addition, APDE team members have hosted training workshops at NIST on SGML and DTD use specifically targeting the STEP community.

The 10303-210 [184] Development Team was established as an APDE alpha user. As an alpha user, this team received extensive user support including participation from APDE developers in team meetings, pre-release copies of software, and immediate bug fixes to code as needed. In exchange, the team provided valuable feedback on the APDE applications as they were developed and served as a rigorous test case for developing additional APDE applications that use SGML. 10303-210, while still under development, is almost 4000 pages in length. Because of its size and complexity, if it were not for the APDE, it could not have been published; other off-the-shelf proprietary word processing software applications were unable to support it. NIST has received many accolades from the 210 Team and from other APDE users for the quality technical support services such users have received.

9.3.2 Collaborative Partners

The standards development process does not occur in a vacuum. Instead, it relies heavily upon external collaborative partners to provide specialized contributions best met by industry. Industry has generally been slow to provide tools specifically targeted toward STEP standardization due to the small market; however, some members of the commercial community have realized the potential long-term benefit of joining in the STEP development effort early. A concerted effort within the STEP community to attract and maintain collaborative relationships has helped to sustain and gain new interest in the standard as it is progressing.

SC4 has tried to leverage consortium liaisons to gain high (and early) impact on industry's adoption of standards. Formally, several active liaisons have been established between SC4 and each particular consortium. The following liaisons with ISO TC 184/SC4 exist currently:

- European Association of Aerospace Industries (AECMA).
- European Marine STEP Association (EMSA).
- European Process Industries STEP Technical Liaison Executive (EPISTLE).
- Industry Alliance for Interoperability (IAI).
- Object Management Group (OMG).
- Product Data Exchange using STEP, Inc. (PDES Inc.).
- Petrotechnical Open Software Corporation (POSC).
- ProSTEP Association.

Another aspect of consortia support has come through the STEP Centers. These Centers have been established throughout the world and now serve various functions: standards development, training and education, and marketing outreach to the ultimate consumers of the ISO 10303 implementations. Several centers (GOSET, Japan STEP Center [JSTEP]; PDES, Inc., and ProSTEP) are leaders, proactive developers, and validators of several 10303 parts. These same Centers also focus additional energy to customer education and outreach, along with the STEP Centers in Australia, Canada, China, Italy, and Korea.

9.4 CONCLUSION

It is fair to say SC4 has been a unique contribution to the ISO community. It has brought to the standards development table a wonderful blend of earnest resources, strong industry-driven requirements, and creative administrative processes. As its Secretariat, NIST has tried to keep pace with SC4's needs, surpassing the basic subcommittee support required by ISO. Many of the fruits of labor -- methods, processes, and tools deployed within SC4 -- could be adapted for use by other standards development organizations.

CHAPTER 10

THE FUTURE OF STEP

10.1 STEP DEVELOPMENT

By early 1984 there was widespread recognition across industry as to the importance of sharing product data in digital form among business partners. IGES had been approved as a U.S. national standard, and initial implementations were available from the major CAD vendors. Other national standards efforts had been initiated in France and Germany, underscoring the need for a truly global solution. Therefore, when ISO announced its intent to form a Technical Committee on Industrial Automation, NIST drafted a letter to ANSI suggesting that the committee include an effort on product data standardization. The result was the formation of ISO TC184/SC4. You read in Chapter 3 about the various technical transformations STEP underwent in the years that followed. Chapters 4-9 highlighted some of the technical and administrative innovations the STEP community has contributed. In this chapter, you will revisit a little of the history and the present, as they contribute to forecasting the future.

10.1.1 The Evolution of STEP

The evolution of product data representation was discussed in Chapter 2. This evolutionary process was slow, tedious, and held mixed rewards. Consequently, today's product data exchange environment can still be characterized by the following:

- Use of national standards is still predominant (e.g., IGES).
- Companies are still implementing single system solutions.
- The transition from one product cycle to another is often accompanied by loss of valuable information.
- Paper is still a common vehicle for product information exchange within industry.

In his 1995 publication [185], Julian Fowler characterized the use of product data standards as shown in Table 10-1 (a double "**" indicates the standard that has widest use for each industrial sector):

	IGES	SET	VDA-FS	EDIF	POSC	DXF[36]
Aerospace	**	*				*
Automotive	**	*	*			*
Building and Construction						**
Process Plant	*	*				*
Oil and Gas					**	*
Shipbuilding	*					*
Electrical/Electronic	*	*		*		
Consumer Goods	*					*

Table 10-1: Comparing the Use of Data Exchange Standards

However, there are other, more positive characteristics as well:

- Initial pilot implementations of STEP are in production operations
- Competitive STEP software tools are becoming increasingly available
- STEP is incorporated into major industry business strategies
- The supplier chain is beginning to use STEP

[36] Data Exchange File or Format developed by Autodesk, Inc.

These types of commitments have already exhibited payback for the users. Pilot programs within PDES, Inc. have shown a 10% improvement in reliability of data exchange, a 10% process savings for noncomposite parts, and 50% process savings for composite parts. Eliminating data reentries is another benefit. For instance, savings on tool design for CAD/CAM systems are projected at 27%.

By the year 2000, NIST hopes to see the following:

- No further development of national product data standards
- Single system solutions beginning to diminish
- Paper drawings as a common exchange vehicle for small businesses only
- STEP requirements adopted into new systems procurements
- State-of-the-practice file exchange using STEP for selected business processes
- Emerging shared database implementations in industry and government
- Availability of STEP translators for multiple applications
- Reductions in product development times
- Major cost savings for technical data management

STEP is already a good alternative for many applications today. Table 10-2 shows one view of several such applications [186].

Use of Data	Probably Require Native CAD Format	STEP a Good Alternative
Design	▪ Chassis ▪ Body In White	▪ Lamps ▪ Fasteners ▪ Labels ▪ Switches ▪ Spark Plugs ▪ Filters & Air Controls
Analysis	▪ Finite Element Analysis On CAD-Associated Tool	▪ Finite Element Analysis On Separate Tool ▪ Computational Fluid Dynamics ▪ Noise, Vibration, & Harshness ▪ Crash & Kinematics/Dynamics
Product Data Management	▪ Archival (Legacy Data) & Bill Of Materials	▪ Archival (Legacy Data) ▪ Configuration Management Data Exchange With Suppliers
Manufacturing	▪ Sheet Metal Tooling	▪ Computer Numerical Control ▪ Coordinate Measuring Machine ▪ Fixturing & Rapid Prototyping

Table 10-2: Applications for Using STEP

In five years, companies will assemble widespread STEP-driven manufacturing. A flexible, building-block approach to implementing STEP will be the state-of-the-art. Shared database implementations will be prevalent and the knowledgebase environments you read about earlier will be emerging.

10.1.2. A Modular Approach to STEP

As detailed in Chapter 3, the STEP architecture began with the development of the IPIM (Integrated Product Information Model). The architecture then moved to Context Driven Integrated Models (CDIMs). Today it centers on Application Protocols. The dual history of IGES application subsets covered in Chapter 2, and the use of CDIMs to evaluate the IPIM, eventually reinforced the need for and gave rise to the notion of APs. The capability of sharing the common information defined by two or more APs (Cooperative use of APs[37]) is an important requirement for companies to realize the full benefits of STEP. In the middle 1990s, Application Interpreted Constructs (AICs) were created to provide consistent and standardized interpretation of requirements when such requirements are used across multiple APs. The effect of AICs is to enable cooperative use of APs. This strategic emphasis started in 1994 (See Chapter 4). In the future, companies may be able to choose from a set of comprehensive STEP constructs or modules to satisfy their data exchange needs. It is expected that this "plug and play" environment will emerge prior to the year 2000 (see Figure 10-1).

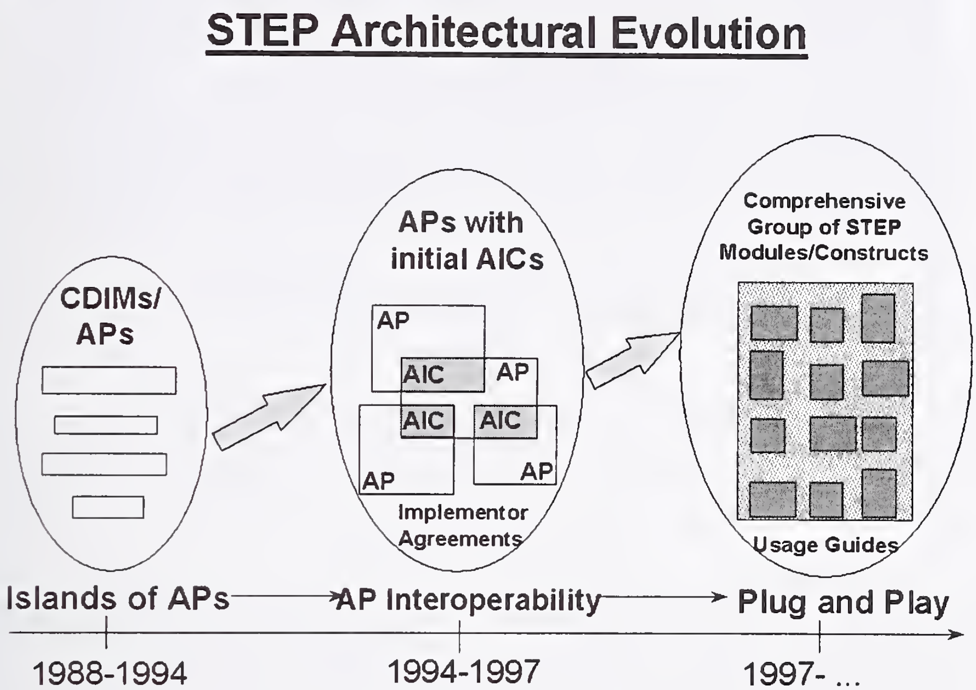


Figure 10-1: STEP Architectural Evolution

To get to this point, a modular extension strategy has been developed, which will enable component-based STEP implementations and reduce the development and publication costs of STEP APs. PDES, Inc., in collaboration with NIST and other STEP Centers, is leading the development of this strategy. It focuses on the harmonization of requirements and their solutions and documenting the result in Application Modules (AMs), which may replace AICs in the current STEP architecture.

Several modular extensions to ISO 10303-203 [187] already identified are Colors, Layer, and Groups; Product Data Management; Drafting; Dimensional Tolerances; and Parametrics. Companies will be able to implement specific modules of functionality to satisfy their business needs. In the future, major AP development work is expected to converge into common modular subsets, as opposed to the current situation where numerous stand-alone APs contain redundant information.

A concept to improve the ISO 10303-standardization process is also in development. This concept allows the construction of extension modules from current APs that are already international standards. These modules will be thoroughly tested through STEPnet, Test Rally, and other testing facilities, and then designated as "advanced industry standards" (see Figure 10-2). Once tested, the modules will be taken through the ISO process more quickly than the current process practiced by SC4. Modules for the Product Data Management (PDM) domain have been

[37] "Interoperability of APs" is often informally used when discussing cooperative use of APs. The intent of either expression is to describe interoperability *of systems* implementing and interacting across multiple APs.

harmonized among the members of PDES, Inc., ProSTEP, and the Japanese STEP Center (JSTEP). This unified PDM schema is the first test case for this concept.

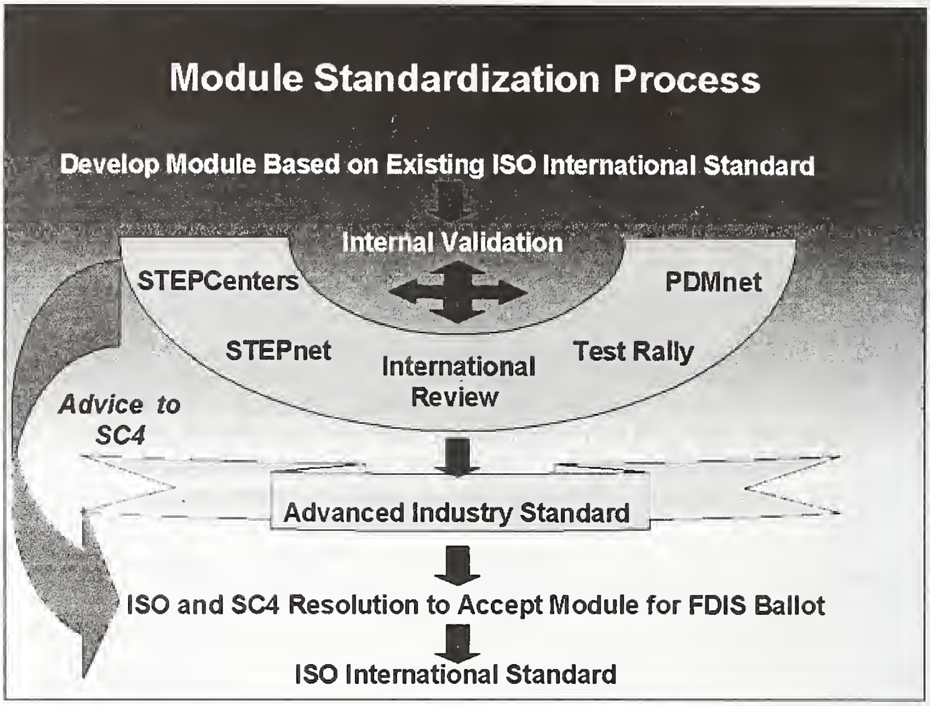


Figure 10-2: Modularization

The initial release of STEP as an international standard in 1994 caused some constraints on making any changes in the architecture of the standard. Increased efficiency in using the standard will be balanced against requirements for making changes in translator software. Current STEP translators already represent a large investment by CAD system developers and users of the standard. Change management of STEP is discussed later in this chapter.

10.2 DATA SHARING

In Chapter 6 you read that several different types of implementations have been envisioned for the practical use of STEP information. Initial work concentrated mainly on the exchange of physical files and the initial release of ISO 10303 only supports file exchange; however, ISO 10303-22 [188] allows database access to STEP users as well.

At a high level, SDAI describes a programming interface to data governed by an EXPRESS schema. Together with standard data definitions, SDAI facilitates the integration of software components from different vendors. The SDAI specification defines the following:

- A programming environment and data dictionary using EXPRESS [189]
- Data manipulation operations, errors, and states
- Implementation classes describing standardized subsets of the specification

The current edition of SDAI is configured for single-user operation; future editions will need to have access control capabilities for multi-user operation.

Future editions of SDAI will also extend existing capabilities in several key areas, one of which is support for industrial data sharing environments with multiple users accessing shared STEP data. Additionally, extensions in the SDAI data dictionary to allow the interface to be used to create EXPRESS schemas have been identified as a priority. SDAI will also be extended to support new developments in EXPRESS and EXPRESS-related languages. Support for the EXPRESS-X [190] mapping capabilities are expected within an SDAI environment, both for producing the mappings and for using the mappings to generate viewing and translation capabilities. Support for the capabilities found in the developing second edition of EXPRESS for defining processes and methods on objects are also expected to significantly expand industrial applications of EXPRESS and SDAI over the coming years. Finally, user requirements and feedback based on industry experiences using SDAI will be examined so that SDAI can be a more powerful and an easier to use interface for programmers developing STEP, or any EXPRESS-driven, implementations.

10.3 EXPRESS

At the heart of STEP is the EXPRESS language. An EXPRESS information model describes the properties required of a data set that stores information. These properties can be structural, such as every entity describing a point shall contain an X coordinate and a Y coordinate, or constraint, such as every loop describing a boundary of a surface shall be closed. Many of the assets to ISO 10303 developers using EXPRESS were highlighted in Chapter 5.

Other SDOs and those outside the standards' development community have created extensive libraries of EXPRESS models. This advantage is key if an organization is going to work on information similar to previously defined information. EXPRESS has been used to define information for mechanical design and manufacturing, electrical design and manufacturing, shipbuilding, and industrial building construction.

EXPRESS has the shortcomings already mentioned in Chapter 5. Also, engineers for engineers have designed it. This has made the language conservative in some respects. For instance, EXPRESS describes constraints using functions and procedures because engineers are more familiar with algorithms rather than the logic-oriented constructs preferred by mathematicians.

At the time of this writing, two efforts are underway to extend EXPRESS. The first is being led as an activity within SC4 for a second edition of EXPRESS [191]. It is working to make minor and major improvements to EXPRESS for use by STEP. Examples of minor extensions being considered include making super and subtype constraints easier to understand, making enumeration and SELECT data types extensible, and allowing generic attributes in abstract entities. More major extensions are being considered to allow EXPRESS to model new kinds of information, for example, business process information.

The second effort is working on a mapping language for EXPRESS. This effort is also being led as an activity within SC4 and the new language is called EXPRESS-X. EXPRESS-X is a language that specifies the relationship between structures in one model and structures in another model. The models that EXPRESS-X addresses are EXPRESS information models, such as the 10303-227 process plant spatial configuration model.

The intent of the new mapping language is to make STEP easier to implement. Relational databases use the SQL language to convert information from a neutral form to an application-specific form. This process is called "defining a view." EXPRESS-X defines views that map STEP data into legacy data and vice versa.

The EXPRESS-X Team is proposing use of EXPRESS-X for at least the following:

- improve the verification of its mapping tables;
- create views of EXPRESS information;
- map legacy data into and out of STEP; and,
- translate information between versions of EXPRESS schemata.

Although EXPRESS-X is not yet a standard, several vendors are developing EXPRESS-X translation systems. Extensive examples are being developed to show how EXPRESS-X can be used to map legacy data into STEP, formalize the definition of mapping tables, and convert data between 10303 parts. The STEP modularization effort is using EXPRESS-X to define the mappings used by its modules.

10.4 UPWARD COMPATIBILITY

Product data is an essential corporate asset. The ability to retrieve, understand, and use product data is a requirement of business. Without that ability, the product data loses its value. The expectation of most users is that committing to the use of STEP will ensure that data will be usable in future years without having to expend significant funds in a data migration effort. Upward compatibility will be a strong consideration during the design of a modification or enhancement to ISO 10303, and must be balanced against the evolutionary requirements and technical integrity of the standard.

The stated goals for STEP noted in Chapter 4, include a requirement that STEP be "upward compatible." While the interpretation of this requirement varies, the concept of upward compatibility usually relates to the effects of differences among a series of successive versions of an implementation. One way to view upward compatibility is to consider the possible effects on new versions:

- As related to the use of data in physical files
- As related to developing and maintaining software translators
- On data access interface programs (e.g., SDAI)
- On application software using STEP data
- On interoperability of APs.

Industry fully expects that STEP will evolve over time; however, the data model forming the foundation of STEP should be a stable, complete, and unambiguous definition of product data. If the definition of the standard is constantly changing, the confidence in STEP as a standard will diminish. Any potential change to that foundation will be examined carefully to assess the impact to existing data (and applications) and to verify that the change is necessary to enhance the stability, completeness, and clarity of the standard. When a change is determined to be necessary, it will be incorporated into the existing baseline in a way that minimizes the impact to existing data and processes. This must be balanced with improving STEP's utility with emerging technology and industry requirements.

The effort devoted to maximize upward compatibility may impact ongoing STEP development projects to better align those ongoing projects with the upward compatibility requirements. Since STEP has moved into production use around the world, others must respect the cost and schedule impacts that excessive change can cause to current users of STEP. The community currently implementing and supporting STEP models would be affected by non-upwardly compatible changes to the standard. It would also affect STEP processor developments and use. Furthermore, the result of changes to the existing standard would delay future products, and divert scarce resources to deal with the issues surrounding multiple versions of implementations. SC4 faces the difficult contradictions of balancing the requirements of the current production implementations of the first three ISO 10303 APs, with that of the 30+ APs currently under development. SC4 does support upward compatibility of STEP, and thus will be providing guidelines and constraints for making modifications to ISO 10303 parts. SC4 is developing policies and strategies for change management that consider the developer, implementor, and user perspectives. (Change Management was introduced in the preceding chapter and is discussed later in Chapter 10.)

10.5 INTEROPERABILITY

Interoperability between APs is a major emerging issue. The aircraft industry, for example, may wish to operate ISO 10303-203 with ISO 10303-209 [192]. There are questions about how this integration can best be done for both file exchange and shared database operation. If the same 10303-21 file contains data conforming to more than one AP, it will be necessary to distinguish entries relating APs to each other. It will also be necessary to determine to which AP each entry relates, and to handle references between entries relating to different APs [193].

At a higher level, a crucial issue is whether a concept as modeled in one AP can be understood in the context of the other AP, provided it is within the scope of both. In some cases the representations may be identical. In others there may be significant differences that require the use of some form of mapping between representations. This will usually be so when the application domains of the APs are significantly different.

In the future, ISO 10303 APs will be built primarily from modular components; however, some requirements for mapping of constructs between APs are likely to remain. Here the EXPRESS-X language may find an important role. It will define mappings between different EXPRESS schemas, and in the first instance between schemas that are closely related. It must be pointed out here that some APs do the same thing in very different ways. For example, ISO 10303-203 provides a means of representing the shape of a part, and so does ISO 10303-224[194]; however, while the first builds up shapes out of elements such as faces and edges, having specified geometry and connected in a particular way, the second defines shapes in terms of form features. A form feature is (in the machining context) a group of faces forming the surface of a configuration such as a slot or a pocket, for which well-known machining strategies exist. Mapping between ISO 10303-203 and -224 will therefore involve determining how to group the ISO 10303-203 faces appropriately into machining features.

Perhaps a more fundamental approach to achieving smooth mappings between components is offered through the definition of ontologies. To date, ISO 10303 has been developed based upon the combined expertise of hundreds of engineers throughout the world, codifying terms familiar to them. These definitions, however, are not stated formally in logically provable forms. Thus, the possibility of ambiguity and misinterpretation exists at all levels of the standard. An ontological foundation will be needed ultimately to address rigorously issues of redundancy and misuse within the standard. A formal ontology will also address another missing piece of STEP: the vocabulary of terms used to populate the defined STEP entities. This problem is not as apparent in the traditional CAD applications of STEP where terms and values have been widely accepted for years (such as Cartesian coordinates for spatial locations). In other areas, the terms are much less certain. For example, the SC4 community has not agreed on the manufacturing process names. Thus, in the draft ISO 10303-213[195], while there may be agreement on the concept of a machining process, one application developer may call a process "mill" while another calls it "milling." Such seemingly trivial mismatches could impede the interoperability that SC4 seeks to achieve with ISO 10303 APs.

Currently, a major driver for architectural change in STEP is interoperability between APs. The issue of interoperability brings up an additional point, which is the tradeoff between extensibility of the specification and guaranteed interoperability of applications using the specification. On the one hand, it would be naive to think STEP developers would have the foresight to anticipate all data elements of importance for any significant time span. For example, the increasing use of parametric design is not yet supported in the current ISO 10303. Therefore, there is definitely a need to be able to expand the current STEP data structures. On the other hand, interoperability between APs is definitely threatened if expanded data structures are added outside the standard.

10.6 CHANGE MANAGEMENT

The SC4 Standard Enhancement and Discrepancy System (SEDS) process was described in the previous chapter. To minimize the need for this process, any proposed standards should be tested thoroughly and validated prior to becoming international standards. Changes to the proposed standard during development and testing phases are to be expected. They are also much easier to accommodate at this stage of a standard's development. Once a standard has achieved international standard status, the community will be much more judicious in making changes. Changes that are not upward compatible should only be considered when the standard is "broken." Broken could mean a technical corrigendum to fix an error, or it could mean an amendment to accommodate future suitability of the existing standard part for use by other developing parts of ISO 10303.

All proposed modifications should include an "impact statement" to the existing baseline, which provides details on the international standards that are impacted. Modifications should also include an analysis of the modification from an upward compatibility and interoperability perspective, and the anticipated benefit prompting the change. All proposed modifications to the STEP integrated resources that will impact upward compatibility of existing implementations should identify the migration path for existing implementations. The migration path could be documented in EXPRESS-X. Proposed changes should be tested with existing APs prior to incorporating such changes into ISO 10303 to remove unwanted or unexpected impacts.

10.7 ARCHIVAL REQUIREMENTS IMPACTING STEP

Archiving technical data in a neutral, public data standard is one of the greatest long-term benefits some companies see for implementing ISO 10303. Archiving data in this way will provide a more time-stable representation than that of a proprietary application's native file format. There are a number of issues about how to ensure the integrity and usefulness of the data over multiple generations of technology. Many companies keep old versions of application systems for accessing data archived in that old version. For that same reason, companies also keep application systems that are no longer part of their product development or maintenance processes. Generally, these types of solutions eliminate the need for converting the data from one format to another, thereby ensuring data that are more complete. When conversion is required, which is most often the case, data can potentially be lost or the accuracy of the data is questionable. In such an instance, the retrieved data may or may not be very usable.

Because of these and other issues, companies are starting to look at neutral formats for archiving their product data. ISO 10303 will fill this need; however, before companies will embrace the use of STEP for archival purposes, more business case data are needed. Companies need some assurance that data archived in the current version of STEP will be compatible with the future releases of STEP and that it will be a cost-effective alternative. A project is underway within PDES, Inc. to test the use of STEP for archival purposes.

In many industries, product data must be available throughout the lifecycle of the product. The need to access the data is due in part to regulatory or legal requirements, product improvement, use on new programs, and development of maintenance processes. The operational life span of some products is very long. For example, the life of a commercial airplane is in excess of 40 years; the life of a ship is in excess of 50 years. Digital product data must be accessible and interpretable decades after it was developed and stored initially. The rapid change in information technologies makes hardware and software systems obsolete in just a few years, so the systems trying to read the data may be very different from the systems that created it. Even an international standard for data will evolve over time and this must be considered in the retention system.

Design reuse is one of the most important areas that is being impacted by STEP. Companies will be motivated to retain design data in the ISO 10303 format for easy recall by CAD engineers because up to 80% of all new designs of a product are simply redesigns [196]. The ability to rapidly retrieve and update previous designs using ISO 10303 will no doubt reduce costs and time and increase flexibility.

Some fundamental assumptions for using STEP for long-term data retention are included in the following table:

	Fundamental Assumptions for using STEP for Long Term Data Retention
1.	STEP is not going to standardize all company data.
2.	Archived data should be independent of specific application systems (archived data is separate from application software).
3.	STEP file sizes are not going to be a limiting factor.
4.	Retained data must include schema defining the data.
5.	Archived data should be based on an open architecture (e.g., independent of hardware or operating systems).
6.	Other standards will be required for long term retention of complete product data.

	Fundamental Assumptions for using STEP for Long Term Data Retention
7.	Everything in STEP (methodology, technology, EXPRESS, ...) will evolve over time; technologies supporting other layers of the "framework" will also evolve and we need to be able to handle this.

Table 10-3: Using STEP for Long Term Data Retention

It is expected future enhancements to STEP will be needed to support its use as a long-term retention standard. Activities are underway to investigate the impact of the modular STEP architecture on the dictionary requirements of a STEP repository. Also, EXPRESS-X may be used to document the evolution of ISO 10303 parts and provide upward migration to data. Evaluation of the impact of schema evolution on a multi-generational repository (i.e., a data repository containing data corresponding to different versions of schemas) will take place. Finally, evaluating the impact of long term data retention on the STEP implementation architecture, especially 10303-21 and the various SDAI bindings, is planned as part of PDES, Inc.'s agenda.

10.8 ENGINEERING ANALYSIS

STEP is beginning to play a significant role in Engineering Analysis (EA). U.S. companies, such as Boeing and Lockheed Martin, are taking the lead to create an interoperating suite of EA APs. These protocols leverage both those ISO 10303 APs developed to date, and the ESPRIT Generic Engineering-analysis Model (GEM) [197].

The initial effort to execute the Engineering Analysis activity within ISO TC184/SC4 will be to define the Engineering Analysis Core Model (EACM). The EACM will then be integrated with the existing ISO 10303 integrated resources, integrated application resources, and the APs that concern shape and engineering analysis. This effort includes harmonizing the representation of product structure, shape, composites, and unstructured-grid finite element analysis. The goal is to ensure that the EACM maps completely to the ISO 10303-209 [198] ARM. This will help ensure that any new EA APs in this suite will interoperate with 10303-209 and the work that has already been done to harmonize 10303-209 with 10303-202 and 10303-203. The common information requirements will be expressed as Units of Functionality that may be included within one or more new APs as required. Additional analysis disciplines such as kinematics, materials services, and systems integration information will be included in the EAC as the project progresses. Three new STEP APs or modules are planned as components of the EA suite:

- Materials Services -- will support the exchange and sharing of information such as elastic and nonelastic material properties, fatigue and fracture characteristics, and allowables for material test specimens and sub-assemblies.
- Aero-thermo/elasticity -- will support the exchange and sharing of information used in simulating the interaction between flight vehicle components and the air. This AP will support analyses such as conceptual fluid dynamics-based aerodynamics and associated thermodynamic analyses using structured and unstructured grids.
- Dynamic Mechanisms Analysis -- will support the exchange and sharing of information used in the dynamic simulation of mechanisms with flexible links.

A related effort, Electro-Mechanical Sub-Systems, will support information such as control-law representation, mechanism analysis, and state analysis definitions.

The following components will be used in initiating the development of this suite:

- ISO 10303 integrated resources, parts 104 (FEA) [199] and 105 (Kinematics) [200]
- ISO 10303-202, -203 and -209
- The GEM model [201]
- The Boeing/U.S. Navy, David Taylor DT - Nurbs mathematical function representation EXPRESS model [202]

- The U.S. National Aeronautics and Space Administration (NASA) Complex Geometry Navier Stokes Computational Fluid Dynamics Model [203]

The business case for the Engineering Analysis project has been built upon the Boeing Product Simulation Integration (PSI) Initiative, the Lockheed Martin Modeling and Simulation program, and the NASA/Lewis Turbine Aeroelastic Analysis project.

10.9 DESIGN INTENT AND PARAMETRICS

The communication of design intent is very important for companies involved in design activities. Since up to 80% of design tasks are to adapt an existing basic design to new requirements, knowledge of the original design intent is crucial to achieve cost savings.

The Parametrics group in ISO TC184/SC4 is currently developing new ISO 10303 IRs providing capabilities for the representation of product models whose definitions include parametrized dimensions, geometric constraints, and form features. All modern CAD systems generate models with such characteristics, but their development occurred too late for the associated data to be included in the initial release of ISO 10303. Significant technical problems are being encountered in achieving these new requirements, but useful progress is being made. It is expected that ISO 10303-42 [204] resources will continue to be used for the representation of "static" product models. The new resources will supplement ISO 10303-42 by providing "dynamic" capabilities for modeling parts that can be modified by editing dimensional parameter values subject to constraints imposed by the designer. Development of these new resources is led by NIST, and is the short-term component of the work of the Parametrics Group within TC184/SC4. Once the ability to represent parametrized models becomes available, it will be used in future versions of many of the STEP APs.

The Parametrics Group also has a "long-term" activity, which is intended to address aspects of product definition that are less well understood than those covered by STEP as it currently exists. The three topics within its scope are

- Capture of design rationale.
- Representation of design evolution.
- Knowledge representation.

The first deals with reasons for design decisions made during the creation of the product model. The second, regarded as a dynamic process, covers the accumulation of increasingly detailed design information from the earliest stages through to the final detailed design. The third aims at capturing the knowledgebase that constrains and guides the design during this process. None of these capabilities is available in the majority of mainstream CAD systems, though there are signs that their provision will become widespread in the future. The long-term activity of the ISO TC184/SC4 Parametrics Group is to track these capabilities as they develop, and to provide a timely means of capturing their information in a standard form when this becomes appropriate.

There is little published research in the areas covered by the Parametrics work. The ENGEN program (Enabling Next Generation Mechanical Design), sponsored by the U.S. Defense Advanced Rsearch Projects Agency (DARPA) and PDES, Inc., has recently provided a technology demonstration of the capture of some key aspects of design intent through a limited focus on parameters, geometric constraints, and features [205]. This project has been working closely with the ISO TC 184/SC4/WG12 Parametrics Group, and the ENGEN Data Model (EDM) is influencing the design of the new IRs mentioned above.

Other papers [206, 207] address the wider concept of design rationale, the information explicitly recording the design activity and the reasons for choosing design alternatives. The integration of such information with the design intent structures currently under development will provide a powerful capability for the transmission and archiving of truly comprehensive design data in the future.

In addition to the new parametrics IRs, two related STEP modules have recently been drafted that cover mathematical expressions and geometric constraints. These are part of an intended demonstration of the proposed new modular architecture of STEP.

10.10 STANDARD PARTS

Any product design contains concepts that were developed in prior design activities. Furthermore, the product that is designed today will likely be incorporated in new designs tomorrow. This concept of design reuse can be thought of as utilizing an existing design as a "part" or "module" of another design.

Standard parts are design objects chosen for frequent reuse in other implementing designs because they have already proven themselves in operational use.[38] Henry Ford, with his revolutionary concept of interchangeable parts, seems to have been one of the first people to think of "standard parts." These do not have to be limited to fasteners and other relatively simple mechanical objects. A complex mechanical assembly can be considered as a standard part of a higher level mechanical assembly. An example of this is an automobile engine or transmission where the design in its entirety is used in more than one automobile.

In electronics, standard parts are more than resistors, capacitors, or transistors. A personal computer often contains a motherboard that provides interface capability to a large number of other printed circuit assemblies. The motherboard is a standard part from the viewpoint of the computer developer -- there is no need to design a new motherboard for each new computer. Doing that would be counterproductive to the concept of utilizing the expansion of the PC that the motherboard allows.

A library is thought of as a repository of information about standard parts (e.g., a library in a CAD system is the place where a user may get configuration-controlled, approved concepts for use in design). These library items may be functional models, such as of a microprocessor circuit, or physical models, such as the microprocessor functionality packaged in any one of a variety of mechanical packages. There should be no limit to the information available about a standard part -- if the information exists, the information should be available.

The Parts Library standard (ISO 13584) is under development in ISO TC184/SC4. A few parts of the standard have already become international standards. Its purpose is to enable libraries of parts to be accessed in a uniform manner by designers. Standardized parts libraries will be critical to product model data exchanges because they will allow organizations to exchange data within a large system. The shipbuilding industry is particularly interested in this standard and feels that they will not be able to exchange ship data without it. The intention is that this standard interoperate with ISO 10303, but there have been some problems in achieving this in the past. Some of the issues contributing to this inability to interoperate are:

- The roots of the two standards were developed at different times.
- Schedule restraints (mostly from the customer) prevented time for learning to use existing developments.
- Some existing developments were too old or too slow to adapt to another standard's needs, but, on The other hand, these developments also had to be kept stable for STEP usability.
- There were different customers driving both standards.

Today, the STEP and Parts Library communities of SC4 are working closer to achieve interoperability.

10. 11 ELECTRONICS

Over the last twenty years, electronics have accounted for an ever-increasing percentage of product value. The market for embedded electronic systems highlights this phenomenon. Indeed, their market share as a percent of the

[38] The use of "standard parts" here is not to be confused with the use of "standard parts" to mean international standard parts of ISO 10303.

total electronics market is growing due to the number of products that rely upon embedded electronics to provide the flexibility consumers want. These products range from smart cards that are automating a variety of transactions (i.e., banking, medical) to smart military systems that enable a "fire and forget" paradigm.

A result of these trends is an increasing need to reduce the recurring and non-recurring costs associated with utilizing electronics in larger systems. To accomplish this, firms are applying advanced Electronics Design Automation tools. A variety of standards have emerged to facilitate the application of these tools. These standards have been developed to support the exchange of information between different vendors' tools and to document the Form, Fit, Function, and Interface (F^3I) of the electronic device. Applied with varying degrees of success for over a decade, these standards have developed a large base of entrenched users.

The continuing growth of electronics in products combined with the large number of existing standards within the electronics community will drive the future application of STEP in electronics. To understand the future role of STEP it is necessary to differentiate between the design of systems that use electronics and the design of the electronics themselves.

STEP's primary role will be to support the use of electronics in larger systems. This will be accomplished by providing the mechanisms to document the F^3I of the electronics and to enable sharing this information among domains such as electrical design, mechanical design, and analysis. Coupled with IEC standards, such as VHDL and the Institute for Electrical and Electronic Engineers (IEEE) 1076-93 [208] that define the function of the electronics, STEP APs such as 10303-210 [209] can provide a complete description of the system being developed. The working group on electrical and electronic APs (ISO TC 184/SC4/JWG9) investigated the nature of the IEEE's VHDL to ensure information was not lost because of its very different data representation from that of STEP. Discussions during JWG9 meetings and during the HPS (see Chapter 2) meetings often highlighted the need for an EXPRESS model of the VHDL. NIST funded a study by the University of Cincinnati to explore EXPRESS capability in modeling behavioral languages. The paper [210] from this study recommended an EXPRESS extension to include an entity's temporal attribute data. England and the Netherlands also sponsored early work on the modeling of VHDL.

The exchange of information required to design electronics will also continue to be supported by standards such as EDIF [211], as described in Chapter 2. It has been used historically for this purpose and is still supported widely by the Electronic Design Automation (EDA) vendor community. This plurality of standards utilization will necessitate inter-standard exchange mechanisms. To provide the infrastructure for this, a working group has been organized within the IEC Technical Committee 93[39] to ensure the interoperability of related standards. The approach taken by this working group is to define mappings between EXPRESS models of the proposed standards in question. By defining the relationship between related standards formally, the ability to apply advanced data conversion techniques, such as those described in a paper by Hines and Gadient [212], is enabled. NIST actively participates in IEC TC93 to help facilitate the bridging between these two communities.

Because STEP will provide the mechanisms for inter-domain sharing of product information, STEP will enable collaborative, distributed design processes that will impact the product development process dramatically. The utility of STEP to support concurrent engineering can be seen in A. J. Gadient's paper [213], which documents the benefits of a distributed, collaborative design process. This process uses STEP and describes the role standards played in supporting the concurrent engineering of the engine mount for the Boeing 777 aircraft [214]. The former resulted in Lockheed Martin's Center of Excellence for printed circuit assembly being identified for a U.S. manufacturing best practice award [215].

In the longer-term, the ability enabled by STEP to share and apply knowledge in an automated concurrent engineering environment [216][217] will produce revolutionary changes in the way future electromechanical products are designed, manufactured, and maintained.

[39] IEC TC 93: Design Automation

The electronics industry is noted for creating new technologies. Mechanical applications are incorporating more electronics in their designs, resulting in complex, electromechanical products. Today's electrical and mechanical CAD systems are not interoperable, creating a significant barrier to reducing product development cycle time and improving engineering productivity. Multifunction design efforts for products will benefit from concurrent engineering or integrated product design. In the engineering disciplines related to electronic product design, the design process and the inability to exchange product data handicap firms when trying to implement concurrent engineering. STEP provides a means for developing tools that enable interoperation among different systems and for exchanging product data.

Today's highly skilled designer is handcuffed by the inability to share and interpret data outside of a specialized domain or even outside of a proprietary data set. In the future, STEP will play a key role in moving toward an innovative, implementation-independent architecture to provide seamless data sharing, real-time access to cross-disciplinary component libraries, and configuration management capabilities. This will enable design collaboration for creating world-class electromechanical products.

10.12 SUPPLY CHAIN

A technology assessment of manufacturing applications by the Gartner Group [218] identified the era from 1967 until 1997 as that of Manufacturing Resource Planning (MRP). The era from 1995 to 2005 is identified as the "Supply Chain Era." In this era, companies will operate as virtual enterprises. Virtual manufacturers will evaluate applications and application architectures that enable rapid coupling and de-coupling of business processes and the ability to work within a heterogeneous environment. The current AutoSTEP project mentioned in Chapter 8 is examining interoperability across the automotive supply chain. Table 10-4 [219] shows those suppliers participating in this project and the real-life product data exchanges necessary across dissimilar CAD systems.

	Allied Signal	**Dana**	**Eaton**	**Saginaw**	**TRW***	**SPX***
Chrysler Catia	System: *ProE* To/From: Catia		System: *UG* To/From: *Catia*	System: *CTA* To/From: Catia	System: *CTA* To/From: *UG* (SPX)	System: *UG* To/From: *Catia* (TRW)
Ford *CV* *Aries*	System: *ProE* To/From: *CV* *Aries*					
GM *UG* *Catia*		System: *Catia* To/From: *UG*				

* TRW & SPX participated under the oversight of Chrysler. SPX is a supplier to TRW; TRW is a supplier to Chrysler.

Table 10-4: Supply Chain Product Data Exchanges

Large manufacturing organizations are in the process of improving their operations by applying lean and agile manufacturing methods. These methods result in dramatic reductions in manufacturing lead times and, at the same time, being agile enough to change as manufacturing needs evolve. These changes require that suppliers be integrated closer to the manufacturer's MES (Manufacturing Engineering Systems). This situation presents major conflicts to suppliers who achieve part of their cost advantages by having low overhead organizations and provide parts for several manufacturers. For these suppliers to support multiple, high technology interfaces would be a major cost impact and severely erode their cost advantages.

Successfully entering the "Supply Chain Era" has the potential to improve growth and productivity of U.S. industry. Large manufacturers are right-sizing factories through increased outsourcing, and at the same time they are compressing lead times through increased supplier integration. These activities require successful implementation of the capability to not only improve communications with suppliers but to provide feedback and supplier buy-in to manufacturing schedule and cost requirements.

This reality has become apparent to most large manufacturers in the automotive and aerospace industries because they deal with hundreds of suppliers who contribute to the cost of sales by 20 to 70%. When these percentages of costs, which must be controlled by supply chain integration, are evaluated in terms of the $1.5 trillion in sales for all of manufacturing, the possible economic benefit is enormous.

Supply chain integration needs to be enabled by providing communications between customer and suppliers. A current void in this communication is the built-in interrelationship of the technical and business data. Most technical data is controlled outside of the MES that drives the business data. A method for coordinating and establishing relationships among technical and business data can be accomplished by utilizing STEP. Another area affected by this tighter integration is the ability to provide data integrity through its direct tie to the configuration management systems that control changes.

The use of an international, standards-based capability will improve the supplier's ability to compete successfully. Today's manufacturing enterprise is faced with an explosion of new technologies that promise to transform electronic commerce and supply chain management. The last five years have provided an implementation of significant technical capability necessary to support manufacturing supply chains including:

- Electronic Data Interchange (EDI) is increasingly being used to communicate business data in a smart fashion.
- Most major manufacturers are also deploying PDM systems to control their product data.
- The cost of computing has significantly declined such that even the smallest of companies have computers.
- Object technology has developed and is beginning to be widely implemented.
- Agent technology is emerging as a potential to provide configurable interfaces.
- The Internet has also exploded in its use and availability, which enables the transfer of large quantities of digital data across configurable connections. Although the Internet still has security problems, the products to address this security issue are emerging.

Industry is seeing an explosive growth in MES tools, Web access capabilities, and point solutions for managing selected aspects of supply chains. Furthermore, the business environment is quickly moving towards a greater emphasis on global supply chains. Industry response has been to downsize their supplier base, adopting proprietary solutions in order to achieve a tighter coupling. This places suppliers into an even more difficult situation when more than one customer is to be supported.

The economic benefits of achieving supply chain integration are sizable and STEP is an important factor. Supply chain effectiveness is usually measured by cycle time. The capability to reduce cycle time by increasing supply chain integration and agility has the multiple effects of reducing inventories as well as providing the capability to increase market share and price.

10.13 PRODUCT DATA MANAGEMENT

PDM systems are being implemented widely within industry today and PDM systems with STEP functionality are beginning to emerge. A unified STEP PDM module suite (known as the common PDM schema) has been initiated by PDES, Inc., ProSTEP, and JSTEP. The initial version of the schema includes the following units of functionality: item or part identification, authorization, shape, assembly, effectivity, work management, end item identification, document reference, and management resources (security classification, contract, and certification). In developing this PDM module, PDES, Inc., ProSTEP, and JSTEP worked together to harmonize the PDM requirements that are part of several STEP APs, such as 10303-203, -214 [220], and –232 [221]. This PDM module suite has been documented complete with the ARM information requirements and the mapping into AIM using EXPRESS.

PDM systems typically require extensive customization as part of implementation. This is because the PDM systems are implemented to match existing company processes and terminology used within the company. Within the larger companies, PDM systems are being used for teaming and data exchange. Different programs within a company are implementing PDM systems for data access or data mirroring between different program sites. With some of the larger programs where multiple companies are involved, the companies are attempting to utilize PDM systems to manage the data to which the program users have access. In this manner, the users will have the most up-to-date information available. One of the issues related to this is that one PDM system does not necessarily store the data in a format that the different sites or companies can utilize directly. Thus, there is a need for a neutral data exchange standard, i.e., ISO 10303.

The savings opportunities for using STEP are very large in the PDM systems area. STEP is on the leading edge of a paradigm change related to how industry and government are doing business. The cost of computing technology is becoming low enough that small businesses are installing advanced CAx applications as part of the normal process of doing business. With this developing infrastructure, the general industry is able to move into a digital frame of reference and is requesting the data from the prime contractors in a format that is compatible with their respective Commercial Off-the-Shelf (COTS) application. This has many implications:

- Companies that manage their data in a document-based approach are moving to a data management approach that is digital file-based. Thus, this issue is becoming broader (instead of an issue only for bigger business - where big business can develop internal applications to overcome the issue).
- Industry is now able to move to a capability to manage the product definition directly (i.e., digital files containing the product data) versus managing the documentation about the product definition (i.e., drawings).
- Shipping digital files in lieu of hard copy requires that the classic shipping list be replaced with a digital equivalent. In the classic hard copy arena, the sending and receiving systems were human-based.
- Newer designs are more complex and require more complex information to represent the product definition.

Large businesses have been deploying COTS PDM systems to address these issues. A problem with this is that most of the COTS PDM systems require a significant investment in customization to represent the required information. The PDM COTS vendors do not see this as an issue for the larger businesses because of the ability to customize the PDM software for their needs. Unfortunately, this is not the case for smaller business, and PDM vendors are looking for a generic data model that satisfies the data management needs of the classic and newer methods of data management. ISO 10303-203 and -232 have been evaluated by several PDM vendors and seen as a strategic way to have a ready product out of the box that does not require significant customization. Another benefit to the PDM vendors is that if they have a generic data model that is utilized across PDM implementations of their product, the exchange of data between implementations will be significantly easier.

10.14 RAPID PROTOTYPING

The term Rapid Prototyping (RP) can mean different things to different people, and no consensus or standardized definition exists. A working definition which characterizes RP:

- Methods are not necessarily *rapid*, not necessarily *prototyping*.
- It collectively refers to a set of process technologies.
- Physical models and prototype parts are built directly from 3D computer-aided design (CAD) data.
- Parts are built layer-by-layer by joining liquid, powder, or sheet materials using an additive process.
- Layers correspond to horizontal cross-sections of the final part based upon the CAD data.
- Materials can consist of plastic, paper, ceramic, or metal.

Several different systems exist to perform RP processes. These systems are based on a number of different process technologies, with different characteristics for aspects such as materials, performance capabilities, and part sizes. Some typical uses for RP include:

- Communication aids for design reviews and marketing.
- Design verification and evaluation - "form, fit, and function."
- Functional testing and flow analysis.
- Tooling development - master patterns for molded tooling or casting processes.
- Packaging development.
- Medical models - bones, tumors, custom implants or prosthesis.
- Limited production of actual parts.

The current RP industry market has transitioned to two categories of systems: "traditional RP" that focuses on things like accuracy or materials, and concept modelers that focus on aspects like speed and low cost. Some formal standards work has begun. ASTM Subcommittee E28.16 has addressed mechanical testing standards for RP, specifically for tensile strength of RP parts. NIST has also hosted two workshops to initiate discussion on industry needs and requirements for RP standards: STEP-Based Solid Interchange Format (November 1996) and Measurement and Standards Issues in RP (October 1997). The results of the NIST RP Workshop of October 1997 indicated that a CAD/RP data interface standard, Solid Interchange Format (SIF), was necessary. Such a future format would address the shortcomings of existing standards, and enable data transfer for future advanced rapid manufacturing capabilities. Many believe that ISO 10303 may provide existing solutions to satisfy SIF needs. To examine these beliefs more closely, a STEP RP Interest Group formed in SC4 in June 1998. NIST is actively working with collaborators to evaluate the existing ISO 10303 international standards for possible applicability to SIF.

10.15 STEP PRODUCT SUPPORT

10.15.1 Software Tools

In the early 1990s, STEP software development concentrated on software tools to add STEP to existing applications and databases. A growing set of STEP software vendors are providing tools that generate STEP-compatible class libraries, visualize and verify STEP models, and move STEP data into and out of databases. The result of this phase is a set of reliable ISO 10303 translators for CAD/CAM products.

The next phase of STEP software development (late 1990s) is concentrating on tools to integrate STEP data in databases and warehouses. Extensible class libraries will be built on top of SDAI to make it easy for application programmers to find and change STEP data. Many libraries will be made Internet-accessible using the IDL [222] and Java [223] bindings of the SDAI. At the end of this phase, end-users will be able to move information between databases with STEP interfaces. For example, a contractor will be able to find PDM information for a part in the database of a supplier and insert that information into its own database for analysis.

The third phase of STEP software development is several years away. Current developments suggest that it is likely to concentrate on native STEP databases that let large numbers of applications access and manipulate the data of very large products concurrently. For this to be possible, new protocols must be developed to make it easier for STEP applications to share data efficiently and reliably. These protocols may take the form of a High Level SDAI that is layered on top of the basic SDAI to allow applications to share data from the perspective of a chosen AP.

STEP translator quality has increased dramatically over the last two years. Sometimes, however, the models created are of poor quality and therefore translations are not very successful (i.e., bad data in -- bad data out). Some tool developers are working on validation tools to allow the designer to build both model integrity and system interoperability into the model at the outset. These tools have already hit the market, providing significant benefit to the design community.

10.15.2 STEP Translation Centers

Some large companies have established STEP translation centers as a method for effectively exchanging product data among disparate systems. For example, General Motors opened the STEP Translation Center (STC) in Troy, Michigan, in May 1996. The center uses STEP to transfer product designs between different CAD systems. STEP replaces less effective methods of data exchange that have been barriers to streamlining the process of developing new products. The Center is used to exchange designs of new products among GM divisions, their customers, and suppliers. The STC will allow increased cooperation on the design of new products and move them into production in less time and at reduced cost. The initial GM divisions that used the STC to exchange part designs with suppliers were Delphi Automotive Systems, GM Powertrain, and Delco Electronics Corporation.

As STEP translations become increasingly reliable, the need for these centers will diminish and the process will become much more transparent. Small- and medium-sized enterprises (SMEs), however, may still find these centers useful over the next several years.

10.16 INTERFACING WITH OTHER GROUPS AND STANDARDS

10.16.1 Sibling ISO TC 184 Subcommittee

The ISO TC184 subcommittees prepare standards relating to industrial data. Subcommittee 5, Architecture, Communications, and Integrating Frameworks, has a working group, WG1, that deals with modeling and architecture. WG1 has prepared standards about enterprise models, ISO 14258 [224], and enterprise-reference architectures, ISO 15704 [225]. These are high-level standards, focusing on enterprise-level concepts. ISO 10303 usually applies down at the process level where product information is exchanged among engineering and manufacturing applications. When the enterprise *itself* is a product and a project of some organization, then the difference between levels converges. An example of where an enterprise may be a product or project is an architectural, engineering, and construction firm that designs, builds, operates, or disassembles enterprises.

The enterprise (product) must be represented over its entire lifecycle--the scope of ISO 15704. The enterprise model then becomes a product model. In this case, there is considerable opportunity for a STEP AP project to consider shared tools for product representation, for example, using EXPRESS to represent an enterprise model. Some of the enterprise-reference architecture components of ISO 15704 may also prove useful (i.e., reusable enterprise-reference models, enterprise-engineering tools, and applicable enterprise-engineering methodologies).

SC4 and SC5/WG1 have explored areas where they can use each other's technology and standards. The most immediate application is the set of architectural, engineering, and construction application protocols: 10303-221, -225, -227, and -230. Since ships and buildings are similar, the shipbuilding APs also could apply: 10303-215 through -218, and -226. Other areas needing coordination between SC4 and SC5 are ISO 13584 (Parts Libraries), and ISO 15531 (MANDATE). WG1 is planning new standardization work at lower enterprise levels and SC4 and SC5/WG1 anticipate a by-product of that work will point to further coordination opportunities.

10.16.2 Common Object Request Broker Architecture

CORBA is the Common Object Request Broker Architecture developed by the Object Management Group (OMG) [226], a consortium of over 600 members including many software vendors. CORBA defines an integration technology that allows diverse object-oriented applications to exchange data in a ‘conversational’ mode, independent of specific platforms and object implementation techniques.

Every CORBA transaction starts with a client request for information and ends with a server response. In subsequent transactions, the roles of the client and server applications may be reversed. Each information request is routed via an Object Request Broker (ORB) that identifies the appropriate server to provide the required information. The ORB maintains a directory of servers and their services, together with details of their interfaces to the integrated system, both in the client and the server role if appropriate.

In CORBA an object is "an identifiable encapsulated entity that provides one or more services that can be requested by a client." Thus, the application programs in an integrated system are regarded as objects. The IDL language permits interfaces to such objects to be defined for CORBA purposes, independently of the actual implementation of the object. Provision of one form of interoperability between STEP and CORBA is already under way through an IDL binding [227], which is being developed for ISO 10303-22, as discussed in Chapter 6. This will allow a STEP model in a database to be treated as a server in a CORBA implementation.

The types of "objects" dealt with by CORBA exist at the level of entire models or files from the STEP point of view, rather than at the level of individual entities within models. Thus, the primary relationship between STEP and CORBA will be in the area of PDM. Resolution of incompatibilities between the STEP and PDM approaches to handling product configuration management data may have some influence on the future development of the STEP architecture. SC4 and OMG standard developers are actively working toward harmonizing the way PDM data is handled.

10.16.3 Internet and Intranet

The Internet and the Intranet are playing a larger and larger role in the daily lives of the average American, as well as the average engineer. The Internet is being used widely for access to data and information on a global basis. Many commercial companies are using the Internet for advertising their products and providing catalog information on their products. This media provides standard part suppliers, as well as custom design businesses, an opportunity to advertise and make their product available to a broader market with little or no additional cost.

The Intranet use within companies and organizations is broadening because of the ready access to data in a format that is compatible with Hypertext Markup Language (HTML) browsers. It provides company access to such information as standard part data, data or drawing viewers, release information, and status in a guaranteed secure fashion.

The marriage of STEP and the Internet offers some exciting prospects for the future of STEP. STEP thrives in a networked implementation environment. Unfortunately, the early STEP implementations were specified before the Internet explosion. The 10303-21 text file exchange, with its dependency on special-purpose parsers, has not easily found a home on the Internet. The SDAI is a single user data access interface for STEP-based applications; it is not designed to support networked applications. OMG's CORBA may provide some of the tools to bring STEP to the Internet. For example, CORBA promises location transparency for STEP models across an ORB-enabled Internet. With CORBA it will be possible to find a STEP model without knowing its precise location; however, the CORBA distributed object paradigm is not designed to support the STEP requirement for moving data objects from one location to another. Thus, while CORBA may help to bring STEP to the Internet, it is not sufficient in itself.

The future of STEP on the Internet depends on our ability to define an effective integration of STEP and Java. A project within ISO TC184/SC4 is examining this challenge. The goal of the project is to make EXPRESS-based [228] data objects as accessible on the Internet as HTML objects. The Java pass-by-value paradigm is the enabler that will make this possible. A Java-STEP Internet will rely only on proven Internet technologies to work: Hypertext Transfer Protocol, HTML, Java, and Java Object Serialization. With Uniform Resource Locators as persistent identifiers for STEP and EXPRESS-based data objects, the Internet becomes a worldwide repository for shared product data. Moreover, the Internet also becomes the clearinghouse for libraries of STEP EXPRESS classes. With Java's "write once, run anywhere" potential, these classes can be downloaded from the Internet and executed anywhere. Java is a key to the popularization of STEP. There are currently over 400,000 practicing Java programmers who are producing new Internet applications at an amazing rate. The Java programming environment promises to provide even greater programming productivity.

PDES, Inc., in collaboration with NIST, proved through the United States Air Force PAS-C Program demonstrations that data access through HTML viewers is viable and usable through fairly inexpensive products (e.g., Netscape). The data structures within 10303-232 for top down breakdown (i.e., part, document, or mixed) and file information for the top down breakdown is defined for that capability to be exercised. 10303-232 also provides the data structures to provide catalog information for products with part family and part classification information. A populated 10303-232 data file provides a capability, when the instantiated data is converted to HTML format, to provide users relatively inexpensive access to data without expensive CAx applications. HTML does not provide data structures for a data model, but provides an ability to relate different 'existing' data together.

The extensible Markup Language (XML) can enable product data exchange as an alternative to the existing ISO 10303-21 as an encoding of the STEP schema instance. Although XML will probably be most suitable for exchange of product data that is not geometry-intensive (e.g., change orders) and where exchange files are not overly large, it enables WEB-based distributed PDE implementations.

XML is a standard being developed under the auspices of the World Wide Web Consortium (W3C). It is a format for structured data interchange over the Internet, and most Internet browser vendors plan to support XML. Why is XML important to industry and to STEP? It

- Supports data exchange between heterogeneous systems.
- Data sharing between manufacturing applications and common business software tools.
- Facilitates electronic commerce.
- Reduces start up costs.
- Enables interoperability between different transaction processing systems.
- Enables seamless integration between Internet and desktop.
- Provides an on-ramp to the Internet for data represented using standards developed prior to the ascendancy of the Web, such as EDI or STEP.

Why XML instead of HTML? Well, XML is extensible (content providers can develop their own tag sets); HTML is not. XML documents must be either valid with respect to a document type definition, or they must be well-formed; HTML documents may contain tagging errors. XML is designed for representing structure; HTML is designed mainly for presentation. XML documents are intended for interpretation by applications (after being processed by a parser); HTML documents are intended to be read by humans.

10.16.4 Electronic Data Interchange (EDI)

EDI is the exchange of business data between trading partners, as defined by the ANSI-Accredited Standards Committee (ASC) X12 standard [229] in the U.S. or by the United Nations Electronic Data Interchange for Administration, Commerce and Transport United Nations (EDIFACT/UN) standard. Since it is often required to associate product data with business data, it is clearly desirable for STEP to interoperate with EDI.

A recent project performed for NIST and the U.S. CALS Office studied the requirements for EDI and STEP to work together. A combination of the two types of data can be considered the components of a `technical data package'. Initially, DoD technical data packages were examined to establish the scope of the required items of information and their interrelationships. Short, medium, and long-term strategies were defined for achieving interoperability. The short-term solution relied mainly on the capabilities of EDI, but work on the medium-term solution is already under way within ISO TC184/SC4 in developing 10303-232.

STEP and EDIFACT are viewed as complementary standards, addressing different applications in the field of electronic commerce. It is important to note that harmonization between the two standards will not necessarily lead to any modification of any of the standards. In the long range, harmonized definitions of concepts, through the use of a single dictionary, are envisioned. SC4 is actively participating in a work effort, under a common Memorandum of Understanding between ISO and IEC Central Secretariats, and several other ISO and IEC technical committees to align these two efforts.

10.16.5 Object Linking and Embedding (OLE)

Object Linking and Embedding (OLE) is based upon the Component Object Model (COM) jointly developed by Microsoft and DEC. In effect, OLE provides a mechanism for constructing compound documents (in a generalized sense), regarded as objects, and COM is the associated means for communication among distributed objects. Currently, OLE/COM is only available under Microsoft Windows and Windows NT. The expectation is that it could become a de facto standard, at least for PC-based applications. CORBA, as earlier described, has been developed by the rest of the software industry to serve much the same purpose as COM, and various options exist for an associated compound document format.

Product lifecycle application software CAD vendors are beginning to migrate to the use of PC platforms. If vendors uniformly continue this migration, it may become very important for ISO 10303 implementations to interoperate with OLE and COM. This interoperability is for the same reasons that interoperating with CORBA is a current requirement. Significantly, an extension to OLE is currently under development, known as OLE for DM, where DM signifies Design and Modeling [230]. At present, this seems to be restricted to handling the spatial arrangements of graphical objects, but further extensions could bring OLE for DM closer to the scope of STEP and increase the need for interoperability. Currently, however, there is no work in progress towards this end.

10.16.6 Java

Java [231] is a platform-independent, object-oriented programming language developed by Sun Microsystems. It allows a program to be written once and then run anywhere on the Internet. Interoperation of STEP with Java could involve, for example, the use of Java programs for visualizing STEP models over the Internet. Originally the Java language specification was submitted to ISO/IEC JTC1 for publication as an international standard using the Publicly Available Specification process. Now, Sun has opted to pursue the technology's standardization first through the European Computer Manufacturers' Association, rather than via JTC1. As highlighted in Chapter 6, work has already started within ISO TC184/SC4 on a Java binding to the STEP SDAI.

10.16.7 Mandate

The Manufacturing Management Data (MANDATE) standard (ISO 15531) is intended to cover standardized representations of manufacturing information other than product-related data. This includes such topics as manufacturing resources, materials flow, and managing manufacturing. Work on MANDATE is at a relatively early stage in SC4. When detailed MANDATE models are created in any of the relevant application areas, it is envisioned they will draw upon ISO 10303 resources and be designed for interoperability with ISO 10303 application protocols.

10.16.8 Standard Generalized Markup Language

SGML is the Standard Generalized Markup Language [232], an ISO/IEC standard for computer-based documentation. It has already been suggested that a close connection should be created between this standard and STEP since much of the information in a product description consists of textual documents. The aim is to enable creating structures in which SGML documents can be embedded in STEP files, with appropriate references between EXPRESS-based product information and the SGML documents, and vice versa. This will lead to a major and very desirable expansion of the STEP concept of "product representation."

An added effect of interoperability of SGML with STEP will be that the EXPRESS string, currently a non-computer-interpretable data type, will gain intelligence. Thus, in principle, SGML strings transmitted in a 10303-21 file could be subjected to a further level of interpretation after postprocessing of the file. It is significant that strings defined by any other ISO standard are also valid SGML strings, so that these computer-interpretable strings could include (for example) code segments in programming languages such as Ada [233], C, or FORTRAN. The embedding of EXPRESS strings in 10303-21 files would also become valid, and applications for this capability have already been identified.

10.16.9 Virtual Reality Modeling Language

Virtual Reality Modeling Language [234] is sometimes mentioned as an 'alternative' to STEP. In fact, the two have very little in common. VRML was developed primarily for creating interactive 3D simulations on the World Wide Web. It provides purely graphical capabilities, and has no provision for representing non-shape-related engineering data. The language provides several Constructive Solid Geometry-type primitive shapes for visualization, but shapes that are more complex must be represented by polyhedral approximations.

VRML is in the public domain. It is being developed as an ISO/IEC JTC1 standard in collaboration with the VRML Consortium, and is based on the Open Inventor modeling format developed by Silicon Graphics Inc. There may be virtue to provide a means for translating STEP shape models into VRML models for visualization on the World Wide Web; however, this appears to be the only form of interoperability that is likely to be useful.

10.17 CONCLUSION

ISO 10303 technology is causing significant cost savings, higher quality, and reduced time-to-market for companies around the world. It is becoming a major building block in our global economy. STEP is being used to unite manufacturing efforts among corporate partners, distant suppliers, and across diverse computer environments. It is becoming apparent that STEP is much more than an international standard for exchanging product data. It is about enterprise integration, global competitiveness, data archiving, design reuse, and solving challenging manufacturing and business problems. Yet there is so much more to be done. The broadening application of STEP in the twenty-first century holds exciting opportunities!

CHAPTER 11

EPILOGUE

STEP has been, and continues to be, an important standard and technology effort for NIST. It exemplifies how the NIST staff works with industry and standards development organizations to convert industry needs into standards to meet those needs.

The STEP standardization initiative was a unique effort – an experiment. It did not simply take existing commercial applications and choose one or a modification of several, as a standard. Rather it first involved advancing the state-of-the-art in product data technology and then building a standard to meet the emerging vendor capabilities in the new technology. The STEP effort has been successful because it has been driven by the user community and not just by the vendors. It was also particularly innovative within ISO because it:

- Built a close liaison with many research projects worldwide.
- Was well supported by a network of national STEP Centers aligned with industry in their respective countries.
- Incorporated feedback from parallel implementatation activities to build a better standard.
- Developed software tools for building and checking the standard.
- Developed software tools for building and checking implementations of the standard.

It is important to credit the STEP initiative participants with many pioneering accomplishments that were noted throughout this text.

Many of the AP projects employed the paradigm of consortia building the standard to meet their requirements. Prior to maturing the developing AP standard, the consortia usually implemented pilot projects to determine the usefulness and correctness of the proposed specification.

NIST has been involved in every aspect of STEP activities. We have served in leadership roles in both the national and international standards development organizations. We have done research with industry partners on developing some of the underlying STEP technologies. We have been an active member of the PDES, Inc. and PlantSTEP consortia where most of the major aerospace, automotive, and process plant U.S. companies have invested their resources in STEP. We have been the prime U.S. focal point in developing the concepts and the conformance testing methods for ISO 10303 APs.

In 1998, NIST initiated the process of relinquishing ANSI's representative role for the ISO TC 184/SC4 Secretariat. The following letter from the SC4 Secretary to ANSI announced our intentions and changing role:

December 4, 1998

Mr. Kevin Sullivan
American National Standards Institute
11 West 42nd Street, 13th Floor
New York, New York 10036

Dear Mr. Sullivan:

This is to formally notify you that in an ongoing effort to maximize the use of our resources to contribute technically to manufacturing system integration standards, supporting tools, and methods, the National Institute of Standards and Technology (NIST) wishes to transition its role as Secretariat of ISO TC 184/ SC4 to another organization. We would like to complete the transition by 1 October 199 at the latest...

NIST has served in the capacity of Secretariat, on behalf of ANSI, for approximately fifteen years, and we are proud of the legacy we have created during our service as Secretariat of SC4. We moved a fledgling idea from four people at the first meeting in the early 1980s to a multi-national, multi-million-dollar initiative. From this leadership position, we have advanced to initial release, several parts of two international standards: ISO 10303 (most commonly known as "STEP"), and ISO 13584, a standard for Parts Library. Both standards continue to mature, and there are two other standards initiatives underway within the SC4 community. More than 250 people gather every four months to continue developing SC4 standards. STEP, the most well known of the SC4 standards, has committed implementation from all top ten CAD/CAM vendors, and committed production use from both the automotive and aerospace industries here in the U.S.

Our assessment is that the goal of creating and maintaining international momentum in support of these critical standards has been achieved, and this momentum is now self-sustaining. NIST intends to remain strongly involved in the ongoing technical work of SC4 to address the critical industry need for development of testing methods and technologies underlying the establishment of conformance certification programs for these families of standards.

...NIST has developed software tools and services supporting the infrastructure of the SC4 standards-making community. We are willing to negotiate the extent and length of time of our continued provision of such services with the next U.S. Secretariat.

Please do not hesitate to ask if NIST can be of service to provide you with any specific information regarding the duties and responsibilities that we have carried out while serving as SC4 Secretariat on behalf of ANSI.

Sincerely,

Ms. Lisa M. Phillips
ISO TC 184/SC4 Secretary
National Institute of Standards & Technology
Building 220 Room A127
Gaithersburg, MD 20899

It has been exciting times for NIST! STEP and product data exchange has come a long way from the late 1970s when the U.S. Air Force Mantech Program first asked NIST to develop a drawing exchange capability (IGES). Experiments with product data exchange & sharing by the Navy Mantech / NIST-sponsored AMRF, developing IGES testing under the CALS sponsorship, the CALS Office sponsorship of the National PDES Testbed, and the Department of Commerce sponsorship of the Systems Integration of Manufacturing Applications (SIMA) program (which continues to this day) have kept NIST in the forefront of the action. The activities in STEP go across many of the operating units (OUs) at NIST -- Standard Reference Data Program; and the Building and Fire Research, Electronics and Electrical Engineering, Manufacturing Engineering, and the Materials Science and Technology Laboratories. These OUs have all actively participated and helped shape the direction of product data standards. Our experience from STEP is now leading us in new directions that include the notion of information technology metrology and the discovery of a science of information-managed manufacturing.

We at NIST consider ourselves fortunate to have worked on such an ambition as STEP. We also look forward to participate in writing the future chapters of STEP, as part of -- the *Grand Experience*.

APPENDIX A

ACRONYMS

The following is an alphabetical list of acronyms used in this publication.

AAM	Application Activity Model
ADM	Associative Data Modelling
AEC	Architectural, Engineering, and Construction
AECMA	European Association of Aerospace Industries
AIAG	Automotive Industry Action Group
AIC	Application Interpreted Construct
AIM	Application Interpreted Model
AIS	Application Interface Specification
AM	Application Modules
ANSI	American National Standards Institute
ARM	Application Reference Model
AP	Application Protocol
APDE	Application Protocol Development Environment
API	Application Programming Interface
APIB	Application Protocol Information Base
ASC	Accredited Standards Committee
ATLAS	Abbreviated Test Language for All Systems
ATS	Abstract Test Suite
AUSDEC	Australian STEP Data Exchange Center
BSI	British Standards Institute
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CALS	Continuous Acquisition and Life Cycle Support
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CATIA	Computer Aided Three-Dimensional Interactive Application
CC	Conformance Class
CD	Committee Draft
CDIM	Context Drive Integrated Models
CE	Concurrent Engineering
CEO	Chief Executive Officer
CFD	Computational Fluid Dynamics
C&G	Complainers and Gripers Committee
CGNS	Complex Geometry Navier Stokes
CIM	Computer Integrated Manufacturing
CIIN	Computer Integrated Information Network
CITIS	Contractor Integrated Technical Information Service
CM	Configuration Management
CODASYL	Conference On Data System Languages
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-The-Shelf
CSG	Constructive Solid Geometry
CSTAR	C-17 STEP Transfer & Retrieval
DARPA	Defense Advanced Research Projects Agency

DBMS	Data Base Management System
DDE	Digital Definition Exchange
DIS	Draft International Standard
DoD	Department of Defense
DPA	Digital Pre-Assembly Process
DSL	Data Specification Language
DTD	Document Type Definition
DXF	Drawing Exchange Format
EA	Engineering Analysis
EAC	Electrical Ad-Hoc Committee
EACM	Engineering Analysis Core Model
EAS	Electrical Applications Subcommittee
EAP	Electronics Automation Program
EDA	Electronic Design Automation
EDI	Electronic Data Interchange
EDIF	Electronic Data Interchange Format
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
EDS	Electronic Data Systems
EIA	Electronic Industries Association
ENGEN	Enabling Next Generation Mechanical Design
EPISTLE	European Process Industries STEP Technical Liaison Executive
ER	Entity Relationship
ERIM	Environmental Research Institute of Michigan
ESPRIT	European Strategic Programme for Research and Development in Information Technology
FDIS	Final Draft International Standard
FEA	Finite Element Analysis
FIPS	Federal Information Processing Standard
GE	General Electric
GEM	Generic Engineering-analysis Model
GKS	Graphic Kernel System
GM	General Motors
GOSET	Operational Group for the Standard for Exchange and Transfer
GPDM	Generic Product Data Model
HPS	Harmonization of Product Data Standards
HTML	Hypertext Markup Language
IAI	Industry Alliance for Interoperability
IAPP	Industrial Automation Planning Panel
ICAM	Integrated Computer Aided Manufacturing
IDEF	ICAM (Integrated Computer Aided Manufacturing) Definition
IDEF0	ICAM Definition 0, a function model
IDEF1	ICAM Definition 1, an information model
IDEFIX	Integration DEFinition IX
IDL	Interface DEFinition Language
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IGES	Initial Graphics Exchange Specification
IPC	Institute for Interconnecting and Packaging Electronic Circuits
IPD	Integrated Product Design
IPIM	Integrated Product Information Model
IPO	IGES/PDES Organization
IR	Integrated Resource
IS	International Standard
ISAP	International STEP Automotive Project

ISO	International Organization for Standardization
IT	Information Technology
ITG	Integration Task Group
JAMA	Japanese Automotive Manufactures Association
JSTEP	Japan STEP Promotion Center
KfK	German Nuclear Research Center
LEP	Layered Electrical Product
LDDT	Logical Database Design Technique
LMTAS	Lockheed Martin Tactical Aircraft Systems
MANDATE	Manufacturing Management Data
MES	Manufacturing Engineering Systems
MOU	Memorandum of Understanding
MRP	Manufacturing Resource Planning
MTs	Mapping Tables
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NBS	National Bureau of Standards
NEDO	National Economic and Development Office (UK)
NIAM	Nijssen Information Analysis Methodologoy
NIDDESC	Navy/Industry Digital Data Exchange Standards Committee
NIIIP	National Industrial Information Infrastructure Protocols
NIPDE	National Initiative for Product Data Exchange
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
OLE	Object Linking and Embedding
OMG	Object Management Group
OODB	Object Oriented Data Base
ORM	Object Role Modelling
OSI	Open Standards Interconnection
PAS	Publicly Available Specification
PAS-C	PDES Application Protocol Suite for Composites
PC	Personal Computer
PCA	Printed Circuit Assembly
PCB	Printed Circuit Board
PDD	Product Definition Data
PDDI	Product Definition Data Interface
PDES	Product Data Exchange using STEP
PDM	Product Data Management
PHIGS	Programmers Hierarchical Interactive Graphic System
POSC	Petrotechnical Open Software Corporation
PPC	Policy and Planning Committee (of SC4)
PSI	Product Simulation Integration
RAMP	Rapid Acquisition of Manufactured Parts
SADT	Structured Analysis & Design Technique
SASIG	STEP Automotive Special Interest Group
SC4	Subcommittee 4 (of ISO TC184)
SCL	STEP Class Library
SDAI	Standard Data Access Interface
SDM	Semantic Database Model
SDO	Standards Development Organization
SDRC	Structural Dynamics Research Corporation
SEDS	SC4 Enhancement and Discrepancy System
SET	Standard D'Echange Et De Transfert
SIG	Special Interest Group

SGML	Standard Generalized Mark-up Language
SME	Society of Manufacturing Engineers
SMEs	Small- and Medium-sized Enterprises
SOLIS	SC4 On-Line Information Service
SPARC	Standards Planning and Requirements Committee
SQL	Structured (or Standard) Query Language
STC	STEP Translation Center
STEP	STandard for the Exchange of Product model data
TAG	Technical Advisory Group
TC184	Technical Committee 184
TISSS	Tester Independent Support Software System
TDP	Technical Data Package
UoF	Unit of Functionality
US Pro	U.S. Product Data Association
US TAG	U.S. Technical Advisory Group
VDA-FS	Verhandes der deutschen Automobilindustrie
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VRML	Virtual Reality Modeling Language
WG	Working Group

APPENDIX B

GLOSSARY OF TERMS

Abstract Test Suite (ATS) - (adapted from [235]) a part of this International Standard that contains the set of abstract test cases necessary for conformance testing of an implementation of an application protocol.

acceptance testing – the process of determining whether a product satisfies predefined acceptance criteria. Acceptance testing is a combination of other types of tests to demonstrate the product meets user requirements.

Application Activity Model (AAM) - a model that describes an application in terms of its processes and information flows [236].

application interpretation - the bringing together of unlike elements, the information requirements of an application context and an information model.

Application Interpreted Construct (AIC) –a logical grouping of interpreted constructs that supports a specific function for the usage of product data across multiple application contexts [237].

Application Interpreted Model (AIM) - an information model that uses the integrated resources necessary to satisfy the information requirements and constraints of an application reference model, within an application protocol [238].

Applicaton Programming Interface (API) – A standard API specifies a mapping between a programming language and the features of a particular service, and thereby provides access to that service from applications written in a particular programming language [239].

Application Protocol (AP) - a part of this International Standard that specifies an application interpreted model satisfying the scope and information requirements for a specific application [240].

Application Protocol Development Environment (APDE) - An integrated suite of software tools to improve quality and increase productivity in preparing STEP application protocols.

Application Reference Model (ARM) - an information model that describes the information requirements and constraints of a specific application context [241].

AutoSTEP – A project designed to support the vision of an automotive industry made up of extended enterprises based on communication processes that hold product and process design and development together. AutoSTEP is demonstrating (piloting) effective product data communication processes in actual use and lay the groundwork for broad deployment. The project is not just demonstrating the technology, but is also building a business case for re-engineered design and development processes that make the best use of the entire supply chain's talents [242].

CAM-I – Consortium for Advanced Manufacturing – International. CAM-I is an international consortium of companies, consultants, and academics that have elected to work cooperatively in a pre-competitive environment to solve problems that are common to the group [243].

CAx – To denote any type of computer-aided system.

computer-sensible - of sufficient semantic precision to permit automated processes to correctly act and interpret.

conceptual model - information requirements in terms of concepts that are specified as formal structures using the syntax of a modeling language [244].

conformance class - a subset of an application protocol for which conformance may be claimed [245].

conformance testing - the testing of a candidate product for the existence of specific characteristics required by a standard in order to determine the extent to which that product is a conforming implementation [246].

construct - a logical grouping of conceptual model elements that conveys a semantic idea [247].

context-driven integrated model (CDIM) – A conceptual information model which represents the information requirements of a discipline or application use. The model is integrated because it draws upon the resources of other models to specify shared information requirements and is not specific to an application [248].

data exchange - the storing, accessing, transferring, and archiving of data [249].

data model – captures the organization and representation of information (e.g., file formats, databases, program data structures) so that it can be used directly in an implementation.

data specification [language] - a set of rules for defining data and their relationships suitable for communication, interpretation, or processing by computers [250].

data retention - Retaining product definition for later use (for reuse or to establish design authority for various requirements); also the ability to read archived data

EDIF – Electronic Design Interchange Format . Originally started in the United States under the auspices of the Electronic Industries Association. Currently, EDIF is sponsored and developed as an IEC standard under the direction of IEC TC93. EDIF is used for the design and exchange of integrated and printed circuit boards.

enterprise system - One that is used company, program, or division wide; and has robust functionality from a product or configuration management data standpoint.

EPISTLE - European Process Industries STEP Technical Liaison Executive. EPISTLE has A-liaison membership to ISO TC 184/SC4. EPISTLE is a forum for the international collaboration of projects and organizations working toward the routine, standards-based sharing and exchange of engineering data in the process and related industries [251].

exchange structure - a computer-interpretable format used for storing, accessing, transferring, and archiving data [252].

EXPRESS-G – a graphical syntax for a subset of EXPRESS.

EXPRESS-V - (EXPRESS Views) is a mapping language invented during a three-demonstration process of validating protocols selected and developed by the National Industrial Information Infrastructure Protocols (NIIIP) Consortium. EXPRESS-V allows one to create alternate representations of EXPRESS models. In the NIIIP project, EXPRESS-V is being used to create an ARM view of the AIM for AP203. EXPRESS-V is an extension of EXPRESS which iterates over instances of a specified entity type to find the one(s) which satisfy a given condition [253].

EXPRESS-X - The goal of the EXPRESS-X language is to define mappings between information models defined in EXPRESS. The EXPRESS-X language allows one to create alternate representations of EXPRESS models and mappings between EXPRESS models and other applications (e.g., IGES). These alternate representations are called views of the original models. The algorithm for deriving the entity types in a view from the entities in an original

EXPRESS model is specified using various types of mapping declarations in the EXPRESS-X language. The EXPRESS-X language evolved from the EXPRESS-V language [254].

IDEF0 (Integration Definition for Function Modeling) - used to produce a "function model". A function model is a structured representation of the functions, activities or processes within the modeled system or subject area [255].

IDEF1 (Integration Definition for Information Modeling) - used to produce an "information model". An information model represents the structure and semantics of information within the modeled system or subject area [256].

IDEF2 - used to produce a "dynamics model." A dynamics model represents the time-varying behavioral characteristics of the modeled system or subject area [257].

IMPPACT – Integrated Modelling of Products and Processes Using Advanced Computer Technologies. ESPRIT II project that developed an early prototype product data sharing environment based on STEP [258].

information model – The requirements for information content and relationships in an implementation-independent way for clear communication among people; understanding what the information is.

integrated resource (IR) - a part of this International Standard that defines a group of resource constructs used as the basis for product data [259].

International Electrotechnical Commission (IEC) - is the international standards and conformity assessment body for all fields of electrotechnology [260].

International Organization for Standardization (ISO) - a worldwide federation of national standards bodies from some 100 countries, one from each country. ISO is a non-governmental organization established in 1947. The mission of ISO is to promote the development of standardization and related activities in the world with a view to facilitating the international exchange of goods and services, and to developing cooperation in the spheres of intellectual, scientific, technological and economic activity [261].

Internet - Originally called ARPANET after the Advanced Research Projects Agency of the U.S. Department of Defense. This electronic network connects the hosts together so that you may go from one web page to another efficiently. The electronic connection began as a government experiment in 1969 with four computers connected together over phone lines. By 1972, universities also had access to what was by then called the Internet.

interoperability testing - the examination of the information exchange and sharing between two specific implementations under test and the ability of each implementation under test to use such information [262].

interoperability testing – the assessment of a product to determine if it will exchange and share information (interoperate) with another product implementing the same specification.

interpretation - the use of resource constructs to specify context-specific relationships and constraints that satisfy application requirements [263].

Intranet - The use of Internet technologies within an organization (or company) to achieve better results than the conventional means of data access and transfer (Intranet has access to Internet but not vice- versa).

IPC – IPC-D-350 is an industry standard from the Institute for interconnecting and Packaging Electronic Circuits. It specifies 80 character, fixed-length record formats to describe printed circuit board products with detail sufficient for tooling, manufacturing, and testing requirements [264].

ISO 10303 - is an ISO International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a mechanism that is capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving [265].

ISO TC 184/SC4 – ISO Technical Committee Industrial Automation Systems and Integration, Subcommittee 4: Industrial Data.

Java – A programming language that developers use to create applets -- small programs that are embedded in Web pages and that run when a user accesses the page or clicks on a certain area [266].

Manufacturing Management Data (MANDATE) - Methods and standardized data which express information exchanged inside industrial manufacturing plants, except for product definition data. The standards being developed under MANDATE are standard parts of ISO 15531.

mapping table – a component of the application protocol, the mapping table documents the traceability of the application information requirements between the specification of these requirements in clause 4 of the AP, and the application interpreted model that documents how standardized constructs are applied to satisfy these requirements in clause 5 [267].

NIAM – Nijssen Information Analysis Method. A graphical data modeling language and methodology [268]. Today NIAM is known as ORM – Object-Role Modeling.

Object Request Broker (ORB) - The ORB provides a mechanism for transparently communicating client requests to target object implementations. The ORB simplifies distributed programming by decoupling the client from the details of the method invocations. This makes client requests appear to be local procedure calls. When a client invokes an operation, the ORB is responsible for finding the object implementation, transparently activating it if necessary, delivering the request to the object, and returning any response to the caller [269].

ORM – see NIAM.

performance testing – the assessment of the performance characteristics of a product such as throughput and response time under various conditions.

product data - a representation of information about a product in a formal manner suitable for communication, interpretation, or processing by human beings or by computers [270].

product data archiving - ([271] paraphrased) the storage of product data, usually long term. STEP is suitable to support the interface to the archive. As in product data sharing, the architectural elements of STEP may be used to support the development of the archived product data itself. Archiving requires that the data conforming to STEP for exchange purposes is kept for use at some other time. This subsequent use may be through either product data exchange or product data sharing.

product data exchange - ([272] modified) the storing, accessing, transferring, and archiving of product data.

product data exchange - ([273] paraphrased) the transfer of product data between a pair of applications. STEP defines the form of the product data that is to be transferred between a pair of applications. Each application holds its own copy of the product data in its own preferred form. The data conforming to STEP is transitory and defined only for the purposes of exchange.

product data management (PDM)/PDM system – A software tool that manages engineering information, and supports managing the product configuration and the product engineering process.

product data sharing - ([274] paraphrased) the access of and operation on a single copy of the same product data by more than one application, potentially simultaneously. STEP is designed to support the interfaces between the single copy of the product data and the applications that share it. The applications do not hold the data in their own preferred forms. The architectural elements of STEP may be used to support the realization of the shared product data itself. The product data of prime interest in this case is the integrated product data and not the portions that are used by the particular product data applications.

The ProSTEP Association - Association for the Advancement and Support of International Product Data Standardization, it was founded in 1993. Hosted in Germany, ProSTEP's members represent a host of multi-national companies. ProSTEP Association represents the interests of its members in developing and introducing ISO 10303 (STEP).

resource - a construct that has been integrated, and is available for use in the specification of context-specific relationships and constraints that satisfy application requirements [275].

resource model – describe aspects of product information such as geometry, tolerances, shape, weight, and size [276].

robustness testing – the assessment of a product to determine how well it performs when supplied data that is difficult to processes, such as, extremely large data sets or data which contain errors.

SC4 On-Line Information Service (SOLIS) – A worldwide publicly accessible service to access the on-line documents of SC4; developing standards for STEP, Oil & Gas, Parts Library, Mandate; working group documentation, supporting tools, national committee and membership information; and administrative data supporting the development of SC4 standards. There are currently two methods to access this data: anonymous ftp (ftp.cme.nist.gov, or 129.6.32.54), and world wide web (http://www.mel.nist.gov/sc4/) .

schema – is an object larger than an entity that defines a scope in which objects are deeclared. Objects in a schema have a related meaning or purpose. Although objects are logically partitioned into groups, the order of the objects in a schema is not important.

schema integration – the integration of information from various models.

secretariat - The Secretariat is responsible for monitoring, reporting, and ensuring active progress of the work of the subcommittee, and shall use its utmost endeavor to bring this work to an early and satisfactory conclusion. These tasks shall be carried out as far as possible by correspondence. The Secretariat is responsible for ensuring that the ISO/IEC Directives and the decisions of Council and the Technical Management Board are followed. The position of the Secretariat is allocated to a national body and this national body shall ensure the provision of technical and administrative services to its respective subcommittee [277].

Standard Enhancement and Discrepancy System (SEDS) – This system and associated procedures identify and resolve issues related to published ISO SC4 documents.

STandard for the Exchange of Product model data (STEP) - the informal name for the international standard, ISO 10303, "Product data representation and exchange."

standard parts - Design objects chosen for frequent reuse in other designs because they have already proven themselves in operational use.

STEP Class Library – A collection of application-independent class definitions used by the application-dependent classes found in the Schema Class Library. The STEP Class Library provides functionality to support a Schema Class Library, a dictionary of the application model, and data files [278].

STEP Center – a nationally designated organization established to further the advance of STEP use within its country. STEP Centers currently exist in Australia, Canada, China, France, Japan, Germany, Italy, United States, and the United Kingdom.

validation - the process of evaluating a system or component to determine whether it satisfies specified requirements [279].

validation testing – the assessment of the underlying specification to which products will be developed. Validation testing attempts to evaluate the completeness, correctness, and consistency of a data model to be used for a standard.

VHDL – The Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) is a formal notation intended for use in all phases of the definition of electronic systems. It supports development, verification, synthesis, and testing of hardware designs; the communication of hardware design data; and the maintenance, modification, and procurement of hardware. VHDL is typically used for top-down system design, full custom chip design, Application Specific Integrated Circuit (ASIC) library development, validation of designs before and after synthesis, and development and debugging of model code [280].

workgroup system - One that is used by a relatively small percentage of employees, tends to be coupled with another system such as a CAD system, and has relatively limited capability.

APPENDIX C

REFERENCES

Chapter 2

Anderl, R. and Schmitt, M. State of the art of interfaces for the exchange of product model data in industrial applications. Institut fur Rechneranwendung in Planung und Konstruktion (RPK, Uni Karlsruhe); 1989 December.

D Appleton Company, Inc. Development Plan for a Product Data Exchange Specification, PDES Version 1.0. Submitted to South Carolina Research Authority 1987 March; revised 1987 July 23.

Encarnacao, Jose L.; Lindner, Rolf; and Schlechtendahl, Ernst G. *Computer Aided Design; Fundamentals and system Architectures.* Springer-Verlag; 1990.

Goldstein, Barbara L. Harmonization of Product Data Standards, or Can I Have My Cake and Eat Yours, Too? Delivered at CALS Expo; 1991 November 12.

Goldstein, Barbara L. Harmonization of Product Data Standards, or You Can Get Too Much of a Good Thing. Delivered at the NCGA Conference; 1991 February 20.

Mayer, Richard J.; Painter, Michael K.; deWitte, Paula S. IDEF Family of Methods for Concurrent Engineering and Business Re-engineering applications. Knowledge Based Systems, Inc., 1993.

O'Connell, Larry. Paragraphs for book on STEP. E-mail to author; 1997 August 11.

Smith, Bradford M.; and Wright, Thomas P.. Guidelines for IGES Application Subsets. Prepared for Department of Defense CALS Program Office; 1987 February 17.

Guide to the IGES Electrical Entities (Pre-cursor to an Application Protocol). *Initial Graphics Exchange Specification Version 4.0.* EACP-2.3; 1989 June 24.

Application Protocol, Initial Graphics Exchange Specification (IGES), Layered Electrical Product. ANS USPRO/IPO-111-1997; 1997 February.

An American National Standard 3D Piping IGES Application Protocol, Version 1.2. ANS USPRO/IPO-110-1994.

ISO Technical Committee 97: Information Processing Systems.: Information processing systems -- Concepts and terminology for the conceptual schema and the information base. Technical Report 90079. International Organization for Standardization; 1987.

Tsichritzis, D.; and Klug, A. (eds.). The ANSI/X3/SPARC DBMS Framework. *Report of Study Group on Data Base Management Systems.*, Montvale, NJ; AFIPS Press; 1977.

Chapter 4

The nature of industrial data--some architectural aspects and issues for SC4. ISO TC 184/SC4/WG10 N31; 1995 October.

Guidelines for integrated resource development. ISO TC 184/SC4/WG10 N31; 1993 September.

Guidelines for application interpreted model development. ISO TC 184/SC4 N532; 19 February 1997.

Guidelines for the development of abstract test suites. ISO TC 184/SC4 N536; 1997 March.

Chapter 5

Schenk, D; and Wilson, P. *Information Modeling: The EXPRESS Way.* Oxford University Press, New York, NY; 1994.

Function Modeling Manual (IDEF0). ICAM Architecture Part II, Volume IV,); Air Force Systems Command, Wright-Patterson Air Force Base, OH; 1982 February.

Information Modeling Manual IDEF Extended (IDEF1X). ICAM Project Report (Priority 6201); Air Force Systems Command, Wright-Patterson Air Force Base, OH; 1985.

Loomis, M. Data Modeling - the IDEF1X Technique. IEEE Conference on Computers and Communications, 1986 March; Phoenix, Arizona; pp. 146-151.

Nijssen, G.; and Halpin, T. *Conceptual Schema and Relational Database Design: A Fact Oriented Approach.*. Englewood Cliffs, NJ: Prentice Hall; 1989.

Chapter 6

Barrett, Myles; Coletta, George; Langenbach, Cliff; Morris, Katherine C.;Mudgett, Chris; and Nieva, Augusto. A High-level Architecture for Implementing a PDES/STEP Data Sharing Environment. PDES, Inc., SCRA, Charleston, SC; 1991 May .

Morris, K.C.; Translating EXPRESS to SQL: A User's Guide. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 4341; 1990 May.

Morris, KC; et al. Validating STEP Application Models at the National PDES Testbed. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 5196; 1993 May.

Chapter 7

Corley, J.; Minutes for the Workshop on Ways to Improve the AP Development Process. Charleston, SC; November 18-19, 1994.

Harrison, R. J.; and Palmer, M.E. Guidelines for the Specification and Validation of IGES Application Protocols. Natl. Inst. Stand & Technol. (U.S) NISTIR 3846; 1989 January.

Kramer, T.R.; Palmer, M.E.; and Barnard Feeney, A. Issues and Recommendations for a STEP Application Protocol Framework. Natl. Inst. Stand. & Technol (U.S.) NISTIR 4755 1992 January.

Liewald, M.H.; and Kennicott, P.R.. Intersystem Data Transfer via IGES. IEEE CG&A; 1982 May; pp. 55-63.

McKay, A.; Bloor, S.; and Owen, J. Application Protocols: a position paper. ISO TC184/SC4/WG4: N1001; 9 September 1993.

Mohrmann, J.; Polly, A.; Speck, H-J.; Wasmer, A.; Corley, J.; and Ryan, S. Proposed extensions to the AP development methodology. ISO TC184/SC4/WG3 N216 (T19); 21 February 1993.

Owen, J.; and Woodall, Alison. IGES Application Protocols - Why and How? Document 0532 445270; 22 December 1989.

Palmer, M.E. Report of SC4 AD HOC Group on Application Protocols. Presentation for ISO TC184/SC4 Meeting, Gothenburg, Sweden; 26 June 1990.

Palmer, M.E.; and Reed, K.A."D Piping IGES Application Protocol, Version 1.1. Natl. Inst. Stand. & Technol (U.S.) NISTIR 4797; 1992 March.

Chapter 8

Bloor, Susan M.; and Jon Owen. Learning Lessons from Conformance Testing. Computer Standards and Interfaces," No. 17; 1995. pp. 231-251.

Mitchell, Mary J.; Initial NIST Testing Policy for STEP — Beta Testing Program for AP203 Implementations. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 5535; 1994.

Kemmerer, Sharon J.; Development Plan: STEP Conformance Testing Service. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 4641; 1991 August.

Chapter 9

Lubell, Joshua. The Application Protocol Information Base World Wide Web Gateway. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 5868; 1996.

ISO TC 184/SC4 Resolutions Affecting the Organization Structure:

1984-07 (# 4) establishes Working Group 1, Technical Coordination and Support, with the following terms of reference:

- To provide technical support and recommendations for SC4
- To coordinate national technical developments
- To resolve technical differences
- To collate national technical contributions for inclusion in the draft ISO document.

1988-12 (# 30): SC4 establishes an editing committee to finalize the text of the Draft Proposal for submission as a Draft International Standard and maintain configuration control so that all future changes be documented and approved.

1990-01 (# 45): SC4 respectfully recommends to the TC184 chair that the title of the SC4 Committee be changed to "Industrial Data and Global Manufacturing Programming Languages."

1990-01 (# 46): Following the assignment of the accepted new work item on a standard parts library, ISO TC184/SC4 establishes a new working group with the following title and task:

TITLE: Standard for the Neutral Representation of Standard Parts

1990-01 (# 48): An advisory group called "Strategic Planning" is to be created to advise the SC4 chair on matters concerning strategic planning and coordination of SC4 activities. SC4 accepts the offer of the member body of France to undertake the convenership of this advisory group.

1990-01 (# 49): An SC4 Project Management Advisory Group shall be established to provide project management and technical strategy for product data work items.

1990-01 (# 52): In response to the request from TC184, and in order to comply with the new ISO/IEC rules, SC4 proposes to reorganize its work on STEP into the following Working Groups:

Resource Models (ex WG1 SG1)

Application Reference Models	(ex WG1 SG2)
Implementation Methods	(ex WG1 SG3)
Conformance & Testing	(ex WG1 SG4)
Framework & Methodology	(ex WG1 SG5/7)
Information Integration	(ex WG1 SG6)

1990-06 (#65): SC4 adopts the reorganization structure as proposed by the ad hoc reorganization group in Reston. The current WG1 is to be disbanded, effective at the end of the Gothenburg meeting, and its current work is to be allocated to the following new working groups:

WG3	Product Modeling
WG4	Qualification & Integration
WG5	STEP Development Methods
WG6	Conformance Testing Procedures
WG7	Implementation Specifications

1990-06 (# 75): SC4 establishes an Editing Committee to finalize the texts of the Parts of the STEP standard for submission as a Draft International Standard and maintain configuration control so that all future changes be documented and approved.

1991-07 (#85): Following the assignment of the accepted new work item on Industrial Manufacturing Management Data (N60), ISO TC184/SC4 establishes a new working group (WG8) with the following title and scope:

Title: Industrial Manufacturing Management Data

1991-07 (# 87): SC4 welcomes the proposal from IEC TC 3 to work jointly on the topic of electrical and electronic product data standards and establishes a joint working group (JWG9) with the following title and scope:

Title: IEC TC 3 - ISO TC184/SC4 joint working group for electrical and electronic applications of ISO 10303.

11991-07 (# 98): SC4 directs its Secretariat to establish an Editing Committee for ISO 10303 in accordance with the ISO Directives. In addition to assistance in the preparation of texts, consistent between themselves and with the ISO directives, the Editing Committee will provide review for technical coherence across texts.

1992-10 (# 130): SC4 requests member bodies to identify and provide qualified people with adequate long-term commitment and funding to support the critical central functions of qualification, integration, editing, training material development, training and configuration management. In addition, SC4 requests PMAG to identify to member bodies the requirements for support for these functions.

1994-10 (# 217): SC4 establishes a new working group to resolve the technical direction and related technical issues of SC4 so that the results are consistent with the SC4 vision and acceptable to SC4 as a whole.

1994-10 (#219): SC4 directs its Secretariat to set up a quality assurance system within the SC4 organization according to TC184/SC4 document N230, Requirements for Future Capabilities of SC4 standards, objective C14: "Establish and implement a quality system for SC4 standards."

1995-03 (# 231): SC4 establishes a Policy and Planning Committee (PPC) to become the single advisory group to assist the SC4 Chairman, Committee, Conveners and Project Leaders:

1995-03 (# 239): SC4 requests its parent committee to revise the title of ISO TC184/SC4 from Industrial Data and Global Manufacturing Programming Languages to Industrial Data.

1995-10 (# 249): SC4 disbands its Strategic Planning Advisory Group and thanks Jean-Pierre Letouzey for his service as Convener.

1995-10 (# 250): SC4 disbands its Project Management Advisory Group and thanks Neal Laurance for his service as Convener.

1995-10 (# 252): SC4 accepts the report of its Policy and Planning Committee. SC4 requests its Secretariat to circulate the attached resolution for letter ballot to SC4 members, and to issue a simultaneous call for a Convener of the proposed new Working Group, experts for this Working Group, a chair of the proposed new Quality Committee and experts for this committee.

1996-06 (#266): SC4 recognizes and supports the efforts of the STEP implementors to collaborate on implementation issues, and encourages the implementors to continue to hold meetings in parallel to SC4 Working Group meetings.

1996-06 (# 270): SC4 resolves to establish a Change Management Advisory function. SC4 further requests the SC4 Chairman to work with the PPC to present appropriate changes to the SC4 handbook to reflect the task of the Change Management Advisory function for Handbook ratification at the next SC4 meeting.

1996-06 (# 272): SC4 resolves to establish a new Working Group 12, SC4 Common Resources. Responsibility for the Application Interpreted Constructs is assigned to this working Group. The Secretariat shall act as the Convener of this Working Group until nominations for convener can be solicited from the P-members and a selection by the Chair and Secretariat ratified by SC4.

Chapter 10

Eastman, C. Out of STEP? Computer Aided Design, 26, (5); 1994. pp. 338-340.

Freund, Kevin. The PDES, Inc. Building Block Approach to AP Interoperability. Proceedings of the ISO TC184/SC4/WG10 Workshop on Architectures for Industrial Data, NIST, Gaithersburg, MD; 7 - 9 January 1997.

Hardwick, M.; et al., Concurrent Engineering with Delta Files. IEEE Computer Graphics and Applications, 15 (1); 1995.

Laurance, N. A High-Level View of STEP. Manufacturing Review, 7 (1); 1994. pp. 39-46.

Staub, Guenter; and Grabowski, Hans. The Challenge of AP Interoperability within STEP., Proceedings of the ISO TC184/SC4/WG10 Workshop on Architectures for Industrial Data, NIST, Gaithersburg, MD; 7 - 9 January 1997.

R.B. Toth Associates. Standardization Needs in Automated Manufacturing. Final report to the National Bureau of Standards; 1986.

APPENDIX D

ABOUT THE AUTHORS...

Anderson, Bill – Advanced Technology Institute (Contributor to Chapter 10) Dr. Anderson has over thirteen years experience in CAD/CAM data exchange, user requirements, geometric algorithm development and STEP model advancement. He has significant experience working with STEP models as a technical team leader of PDES, Inc. (an international consortium whose goal is to accelerate the development and implementation of STEP). Most recently, Dr. Anderson led the ENGEN (Enabling Next Generation Mechanical Design) Program, which is focused on capturing design intent information through the use of parametrics, constraints, features, and construction history. It is jointly sponsored by DARPA and PDES, Inc.

Barnard Feeney, Allison (Author of Chapter 4) Currently serving as TC 184/SC4 Quality Committee Convener and project leader of ISO 10303-302. Past roles include Deputy Convener of TC 184/SC4 WG4, member of ISO 10303-202 development team, editor of ISO 10303-32, author of, or contributor to, SC4 Standing Documents that provide development methodologies for STEP, i.e., 'Guidelines for Application Interpreted Model Development', 'Guidelines for the Development of Mapping Tables', 'Guidelines for Application Interpreted Construct Development', 'Guidelines for the Development and Approval of STEP Application Protocols', 'Supplementary Directives for the Drafting and Presentation of ISO 10303', 'Guidelines for the Development of Abstract Test Suites'.

Bloom, Howard M. (Co-Author Chapter 1, Author Chapter 11) As Division Chief of Manufacturing Systems Integration Division (originally called Factory Automation Systems Division), initiated MEL's efforts in the development of product data exchange standards. Originally managed the activities in the Automated Manufacturing Research Facility program as it related to the use of product data throughout the manufacturing life cycle. Oversaw MEL's efforts in the IGES/PDES Organization, ISO TC184/SC4, PDES Inc. program, the National PDES Testbed Program at NIST, and the STEP development activities with the NIST SIMA program. Personally involved initially as NIST's member of the IPO Steering Committee and the PDES Inc Technical Advisory Committee. Presently involved as the NIST member of the US PRO Board of Directors and the PDES Inc Board. Received Department of Commerce recognition for the NIST leadership in developing the technical basis and direction of the Department's program in product data sharing technology.

Brauner, Kalman - Boeing Commercial Airplane Group (Contributor to Chapter 2) Mr. Brauner served as Chair of US TAG to ISO TC184/SC4. He was a member and presiding officer of the Technical Advisory Committee of PDES, Inc., and was a member of the Interim Technical Committee and the Host Contractor Selection Committee of the PDES Cooperative (now PDES, Inc.). In the IPO, Mr. Brauner was the initial PDES Project Manager; a member of the Advanced Geometry, Curves and Surfaces, Edit, Technical Planning, and Steering Committees; and Chair of the PDES Initiation, authoring two papers that specified requirements and objectives for PDES.

Still earlier activities included: chairing the national committee responsible for formal adoption of IGES as ANS Y14.26M-1981 (Digital Representation for Communication of Product Definition Data); serving as a member of ANSI Subcommittee Y14.26 on Computer-Aid Preparation of Product Definition Data, and being a contributing author of "Digital Representation Physical Object Shapes" which evolved into section 5 of ANS Y14.26M -1981. Mr. Brauner also served as a representative to the CAM-I Sculptured Surfaces and Geometric Modeling Projects.

Burkett, William C. – PDIT, Inc.(Contributor to Chapters 2 and 5, Manuscript Reviewer) William Burkett has over 15 years of experience as an industrial and systems engineer specializing in system analysis and modeling, information system integration, and product data exchange (PDE) technologies. Prior to joining P.D.I.T., he worked for McDonnell-Douglas and Lockheed on PDE technology and standards development programs. Mr. Burkett was a committee chairman within the IGES/PDES Organization from 1983 and a project leader within TC184/SC4 from

1989 through early 1992. He was an active participant in the developing STEP since its inception in 1984, and was responsible for initiating the Quality Assurance aspects of the STEP standard.

Danner, William – Seneca-IT.com (Co-Author of Chapter 3) Bill Danner is President of Seneca-IT.com, an information technology consulting firm. He worked at the National Institute of Standards and Technology for twenty-one years where he conducted information technology research. Among his work in the area of information modeling methods was serving as Convener for the ISO TC 184/SC4 working group responsible for developing the methodology and architecture used in ISO 10303, as well as the specification language EXPRESS.

Denno, Peter – NIST (Co-Author of Chapter 5) Peter Denno has 12 years industrial experience. He joined NIST in 1995. His current work involves STEP and the EXPRESS langauge. Mr. Denno is the developer of Expresso.

Fowler, James E. - NIST (Co-Author to Chapter 6) Mr. Fowler was the Chairman of the IGES/PDES Organization Implementation Specifications Committee and the Deputy Convener of Working Group 7 of ISO TC184/SC4. Mr. Fowler was a primary participant in the initiation and development of the Standard Data Access Specification through 1992. Since that time Mr. Fowler has been involved in the development of manufacturing resource specifications and in management of NIST's Systems Integration for Manufacturing Applications Program.

Frechette, Simon P., NIST (Co-Author to Chapter 8) Mr. Frechette is currently working in the area of manufacturing standards implementation and conformance test development. He has been the NIST liaison to the AutoSTEP Project, and is assisting USPRO to kick off the conformance testing service for ISO 10303 APs.

Goldstein, Barbara L. Maia, NIST (Author of Chapter 2) Ms. Goldstein previously served as: US Delegate to IEC TC93, member of ISO 10303 IPO Electrical Applications Committee, and Chair of the Tools & Technology Council of ANSI Harmonization of Product Data Standards (HPS) Committee. She initiated the STEP Process Plant Application Protocol Planning Project as a consultant to the Process Data eXchange Institute (PDXI). Currently Ms. Goldstein serves as leader of the Infrastructure for Integrated Electronics Manufacturing (IIEM) Project at NIST and Co-Chairs the Factory Information Systems committee within the National Electronics Manufacturing Initiative.

Gruttke, William B. - Northrop Grumman/Team SCRA (Contributor to Chapter 2) Dr. Gruttke has over twenty years CAD/CAM experience in the aerospace industry at McDonnell Douglas (14+ years) and Northrop Grumman (6+ years) (following 10 years of university teaching at St. Louis University). He has been involved in the IGES Committees and the IGES/PDES Organization (IPO) since 1980. He chaired the IGES Advanced Geometry Committee (1981-1986). He was the IPO Chairman from 1995 -1997. Dr. Gruttke led the software development team that developed the format and translators for the Air Force Product Definition Data Interface (PDDI) Program while at McDonnell Douglas. He has been managing the development of STEP AP224 and STEP-driven applications for the DoD Rapid Acquisition of Manufactured Parts (RAMP) Program since 1992.

Hardwick, Martin - STEP Tools, Inc. and Rensselaer Polytechnic Institute (Contributor to Chapter 10) Martin Hardwick is the president of STEP Tools,Inc. and a professor of computer science at Rensselaer Polytechnic Institute. Dr. Hardwick is a member of the Architecture Working Group (WG10) and the Implementation Methods Working Group (WG11) of ISO STEP. He is leading the project to develop the EXPRESS-X mapping language. He belongs to the board of directors of PDES, Inc. and the US Product Data Association. His research interests include engineering database systems, information modeling, and distributed systems and object oriented programming. Hardwick received his bachelor's and doctorate degrees from Bristol University in England in 1978 and 1982. He is a member of the IEEE Computer Society and the ACM.

Jurrens, Kevin – NIST (Contributor to Chapter 10) Mr. Jurrens served as project manager for the NIST Rapid Response Manufacturing (RRM) Intramural Project and Test Team leader for the PDES Inc. Sheet Metal Project. Primary research activities have addressed development of a standardized data structure for representing machine tool and cutting tool data (i.e., "manufacturing resource data"), establishment of a reverse engineering capability for repair and replacement of aircraft components, and development and validation efforts to assess STEP for application to sheet metal die design. Mr. Jurren's current efforts address measurement and standards issues for the rapid prototyping industry, including evaluation of ISO 10303 as an alternative data interface between computer-aided design and rapid prototyping systems. In addition, he continues to lead U.S. efforts to develop and standardize an electronic representation for cutting tool data through the ISO TC29/WG34 standards organization. He served as the technical liaison between ISO TC29/WG34 and ISO TC184/SC4 for representation of cutting tool data. Prior to NIST, Mr. Jurrens worked for Allied-Signal Aerospace, Kansas City Division,.

Mr. Jurrens has authored several technical reports, conference papers, and a book chapter on various manufacturing applications and manufacturing system integration. He has received the U.S. Department of Commerce Bronze Medal Award for superior federal service in 1993 for contributions leading to the acceptance of the STEP standard and in 1997 for developing the manufacturing resource data representation as a baseline standard.

Lubell, Josh – NIST (Contributor to Chapter 11) Mr. Lubell designs and implements software systems for product data exchange applications at the National Institute of Standards and Technology in Gaithersburg, Maryland, USA. His technical interests include STEP, SGML and XML, database technology, and Internet computing. His previous experience includes artificial intelligence systems design and prototyping, software development for the building materials industry, and knowledge engineering.

Kemmerer, Sharon J., NIST (Editor of Manuscript, Co-Author of Chapter 1, Co-Author of Chapter 3, Co-Author of Chapter 9) Sharon was involved with the IGES/PDES Organization and ISO TC 184/SC4 from 1988-1991, primarily in the technical area of conformance testing. She served as Deputy Convener of the ISO TC 184/SC4/WG6, and was a leading contributor to ISO 10303-31 and 10303-32. After a respite from the product data standardization arena, Sharon again returned to the ISO TC 184/SC4 community. This time she served as the Secretary of the SC4 Secretariat from 1995-1997.

Kiggans, Bob – PDES, Inc./Advanced Technology Institute (Contributor to Chapter 10) Mr. Kiggans has over thirty years experience in advanced computer and product data technologies, program and technical management, acquisition management, and command and control systems. As General Manager for PDES, Inc., Mr. Kiggans has responsibility for a widely geographically dispersed technical team.

Kindrick, James D., ITI, Michigan (Co-Author of Chapter 8) Mr. Kindrick has worked for the past seven years on developing and deploying Product Data Information Technology, with significant focus on the ISO STEP standard for product data exchange. Jim's original work within STEP concentrated on conformance testing and standard quality assurance. He is a lead technical design and developer of the NIST/ITI suite of STEP conformance testing tools and techniques. He has since worked closely with both system vendors and users of product data technology to help facilitate the implementation and adoption of STEP within industry. During the past year, he has been an active leader in the harmonization of requirements across STEP APs, primarily focused on developing a common set of PDM requirements and corresponding STEP information models. He has delivered several industry workshops on the topics of PDM and STEP, and is the author of several articles and technical reports. Jim is also an active technical member and administrator of the international PDES Inc./ProSTEP PDM implementation roundtable and interoperability testing forum, focused on integrated PDM and CAD data exchange.

Libes, Don - NIST (Co-Author of Chapter 5, Contributor to Chapter 3) As a computer scientist at NIST, Don has written 16 papers on STEP and is responsible for numerous public domain STEP tools including the first CADDETC-certified EXPRESS toolkit, an EXPRESS-Tcl binding, and the first version of SOLIS. Don also designed and implemented the EXPRESS server and the NICS STEP repository.

Mitchell, Mary - NIST (Contributing reviewer of complete manuscript) Before joining the Advanced Technology Program, Mary Mitchell was the Program Analyst for the Manufacturing Engineering and Information Technology Laboratories and Manufacturing Extension Program in the NIST Program Office (April 1997-1998). Prior to this, she led a group and managed variety of technical projects related to STEP. Mary was one of the contributors to the development and validation of the STEP architecture, including components of the core information resources, application protocols, and conformance testing. She also served as the Deputy Chairman of the IGES/PDES Organization and led one of the WG4 Integration and Qualification teams for a number of years. In February 1998, Ms. Mitchell received the first William J. Conroy Standards Professional Award for her leadership efforts in STEP. In November 1994, Ms. Mitchell received the Department of Commerce Silver Medal Award for her work on STEP. She came to NIST in 1982 to work on distributed database systems for manufacturing, one of the projects within NIST's Advanced Manufacturing Research Facility.

Morris, KC - NIST (Co-Author of Chapter 6, Contributor to Chapter 5) As a computer scientist in the Manufacturing Systems Integration Division of NIST, KC has been a primary contributor to the development and validation of STEP's implementation mechanisms. She has architected and contributed to the development of numerous prototypes of SDAI, as well as,portions of NIST's public-domain STEP Toolset including an EXPRESS to SQL translator (now defunct) and the STEP Class Library. KC has published dozens of articles on the topics and has actively worked with ISO TC 184/SC4's technical working groups and the industrial consortia, PDES, Inc. and NIIIP. Her current research interests include object-oriented and distributed systems for engineering and manufacturing and product data management and techniques for testing such systems.

Nell, James G. – NIST (Contributor to Chapter 10) Mr. Nell is Convenor of ISO TC184 SC5 WG1, Automation Systems and Integration, Modeling and Architecture. He is a member of the US technical-advisory groups to TC184 and SC5. At NIST, he is the principal investigator of the Manufacturing Enterprise Integration Project to evaluate the use of architectures, frameworks, and enterprise models for use by business entities. He was active in the TC 184/SC4 work to develop product- and process-data representation serving as the US expert to the SC4 Strategic Planning Advisory Group for four years. For the IGES/PDES Organization, he chaired the IPO Steering Committee for two years.

Nelson, Paul - Raytheon Systems Company/Hughes Aircraft Company (Contributor to Chapter 10) Mr. Nelson has extensive experience in defense and commercial industries, with an emphasis in electronics and electromechanical fabrication and assembly. Recently, his emphasis has been on CAD-CAM data utilization, machine control instructions, producibility, and standardization of exchange of product model data between engineering and manufacturing. He was a principal architect for STEP AP210 (Electronic assembly, interconnect and packaging design) working with a distributed team of engineers and technologists across North America, Europe, and Asia.

Nicolson, Martha – PDES, Inc./Arthur D. Little, Inc. (Coordinating Editor of Chapter 10) Ms. Nicholson is a Manager with Arthur D. Little and has ten years of experience in program management. Currently she has lead responsibility for providing program management support to the approximately 60 technical resources from the PDES, Inc. member companies.

O'Connell, Larry - Sandia National Laboratories, Retired (Contributor to Chapter 2) Mr. O'Connell Designed Electrical Test Equipment and real-time software for Quality Evaluation Systems Testers at Sandia Laboratories. Chaired the Electrical Applications Committee of the IPO and its predecessor organizations from 1984 -1994. Designed and posted the web page for the Advanced Manufacturing organization of Sandia Laboratories.

Palmer, Mark – NIST (Author of Chapter 7) Mr. Palmer led the development of the first application protocol for U.S. industry, the *3D Piping IGES Application Protocol.* This was a collaborative project with the U.S. Navy and the engineering, process plant, shipbuilding, and CAD/CAE industries. He also led the development of the application protocol methodology for STEP. He is the leader for the NIST project "STEP for the Process Plant Industries" and is the chair for the ISO TC184/SC4 Process Plant Team. Currently, Mr. Palmer is working with representatives of the process plant industries to develop and test a coordinated suite of STEP application protocols that will improve the capabilities and productivity of these industries. This includes managing the NIST Cooperative Research and Development Agreements with the PlantSTEP, Inc. industrial consortium and with pdXi (Process Data Exchange Institute of AIChE). Mr. Palmer is the project leader for the ISO STEP application protocol, ISO 10303-227: Plant Spatial Configuration and is working on developing two additional STEP application protocols, ISO 10303-221: Functional Data and Schematic Representation for Process Plant and ISO 10303-231: Process Design and Process Specifications of Major Equipment.

Parks, Curtis – NIST (Contributor to Chapter 2) Mr. Parks is a Computer Specialist with the Electronic Information Technologies Group of the Electrical and Electronics Engineering Laboratory of NIST. He chairs an IEEE Study Group on Electronic Commerce of Component Information. He was principle author of the Hybrid

Microcircuit Application Protocol for the Navy MicroCIM Program and served as editor to the EAC on the LEP AP. He participated in the initial work of the PDES, Inc. E/E Team and later was the IGES EAC acting Chair. Prior to joining NIST he was with General Dynamics Pomona Division as Weapon Systems Engineer on the Phalanx and Standard Missile projects, and later with the CAD/CAM Engineering Department, he developed electrical CAD applications for the design, documentation and tooling for integrated circuits, printed wiring assemblies, wirewrap boards, cables and harnesses. He was the Division's representative to the IPO, and assumed the responsibility of IGES Change Control Secretary. He was a principle modeler of the electrical reference model in the PDES Initiation Task, the leader of one of the Cal Poly Teams, and a principle member of a second Team.

Phillips, Lisa – NIST (Co-Author of Chapter 9) Lisa Phillips is a Computer Scientist in the Manufacturing Standards Methodology group in the Manufacturing Standards Integration Division of the Manufacturing Engineering Laboratory. Phillips currently serves as the Secretary of ISO TC 184/SC4. Her previous roles have included serving as the Project Manager of the Application Protocol Development Environment project, in which she both developed and led the development of software in support of STEP standards. Her technical efforts in this area have focused on the use of the Standard Generalized Markup Language to advance the preparation and improve the quality of STEP standards.

Pratt, Michael – Rensselaer Polytechnic Institute/NIST (Contributor to Chapter 10) Dr. Pratt is a Senior Research Associate at Rensselaer Polytechnic Institute, Troy, NY. He has spent the past four years on secondment at NIST as a Guest Researcher in the Manufacturing Systems Integration Division. Previously he was Professor of Computer Aided Engineering in the School of Mechanical Engineering at Cranfield University in the UK. He is the leader of the Parametrics Group in ISO TC184/SC4/WG12.

Rando, Thomas - Electric Boat Corporation (Contributor to Chapter 10) Dr. Rando has been employed at Electric Boat Corporation for the past twenty years. He has been actively involved in STEP standards development and a technical representative to PDES, Inc. since 1991. He is the project leader for the C++, IDL and Java bindings to the Standard Data Access Interface. He received his PhD in Computer Science from the University of Connecticut in 1990.

Ray, Steven R. – NIST (Contributor to Chapter 10) Chief of the Manufacturing Systems Integration Division, Manufacturing Engineering Laboratory, National Institute of Standards and Technology (NIST). From a technical specialty in process planning, he co-developed a neutral process plan model called ALPS (A Language for Process Specification), and participated on the development of STEP Part 49 (Process Structure and Properties), on the validation of STEP AP213 (Process Plans for NC Machining), and served as acting Chair of the IGES/PDES Organization from 1989 to 1990. Dr. Ray received his Ph.D. in Mechanical & Aerospace Engineering from Princeton University in 1981, and his Bachelor's degree in Physics from Bristol University, England in 1977.

Smith, Bradford M. – NIST, Retired; Self-Employed (Manuscript Reviewer) Mr. Smith served as Chairman of the IGES Organization from June 1980 and subsequently, Chairman of the IGES/PDES Organization until October 1989. Edited IGES Versions 1-3. Mr. Smith achieved ANSI approval for IGES Version 1.0 then organized the ISO TC 184/SC4 Subcommittee and served as Chairman and Secretary from 1984 to 1995. He chaired the U.S. TAG for ISO TC 184/SC4 from 1984 to 1989. Gained approval for the Initial Release of the first twelve parts of ISO 10303 in 1994.

The Editor would like to extend a special thanks to Al Jones, Dennis Swyt, Lisa Fronczek, Joan Wellington, and Melissa Zeltman for their technical, editorial, and compositional contributions to complete this document. Particular thanks and appreciation go to Beamie Young for the cover design.

BIBLIOGRAPHY

[1] Wilson, Peter R. Draft Report of the C&G Ad Hoc Committee; 1987 November 17.

[2] Ray, Steven R., MSID Presentation to the 1999 Assessment Panel Subpanelists, 1999-02-04.

[3] ISO 10303:1994, *Industrial automation systems and integration – Product data representation and exchange* is the formal name for the international standard familiarly know as STEP.

[4] Bloom, Howard. Technical Program Description: Systems Integration for Manufacturing (SIMA). Natl. Inst. Stand. & Technol. (U.S.) NISTIR 5476; 1994.

[5] Conceptual Schema; ISO TC97/SC5/WG3; 1982 March 15.

[6] Smith, Brad; Nagel, Roger; Wellington, Joan. IGES – Initial Graphics Exchange Specification. Autofact; 1981 November.

[7] CAM-I Geometric Modeling Project Boundary File Design (XBF-2), Report R-81-GM-02.1, Consortium for Advanced Manufacturing, Inc., Bedford, TX; 1981 (Revised 1982).

[8] ISO 10303-42: 1994, *Industrial automation systems and integration – Product data representation and exchange – Part 42: Integrated resources: Geometric and topological representation.*

[9] Application Interface Specification (AIS) (2 vols.), Report R-94-PM-01, Consortium for Advanced Manufacturing, Inc., Bedford, TX; 1994.

[10] ISO 10303-11: 1994, *Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual.*

[11] ANSI: Product Definition Data Interface; Task-I: Evaluation and Verification of ASNI Y14.26M. Booz-Allen and Hamilton Inc.; 1983.

[12] Wilson, Peter R. A View of PDES. General Electric Company; 1987 January 7.

[13] Mason, Howard. The Historical Need for STEP. British Aerospace; undated.

[14] Contribution of West Germany, VDAFS. ISO TC 184/SC4 N3; 1984 June 8.

[15] Mason, H. [13].

[16] Contribution of France, SET: Standard for Exchange and Transfer. ISO TC 184/SC4 N10; 1984 September 25.

[17] Fowler, Julian. *STEP for Data Management Exchange and Sharing*. Great Britain: Technology Appraisals Ltd.; 1995.

[18] Schlechtendahl, Ernst (ed.). Specification of CAD*I Neutral File for CAD Geometry, Version 3.3. Springer-Verlag, Heidelberg; 1988.

[19] Parks, Curtis; McCollough, Roger; Azu, Charles; Savage, Larry; Toomey, Patrick; and Makoski, Thomas. Initial Graphics Exchange Specification Hybrid Microcircuit Application Protocol. Natl. Inst. Stand. & Technol. (U.S.) Tech. Note 1295; 1993 January.

[20] Danner, W.F. A Proposed Integration Framework for STEP (Standard for the Exchange of Product Model Data). Natl. Inst. Stand. & Technol. (U.S.) NISTIR 90-4295; 1990 April.

[21] ISO TC 184 SC4, Minutes of First Meeting; 1984.July 11-13.

[22] PDES Initiation Activities. IGES Organization; May 1986.

[23] Kelly, J.C. Final Report of the PDES Logical Layer Initiation Task; 1986 April.

[24] ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*

[25] Halpin, T. *Conceptual Schema and Relational Database Design.* Prentice Hall; 1995.

[26] Integration definition for information modeling (IDEF1X). Natl. Inst. Stand. & Technol. (U.S.) Fed. Info. Process Stand. Publ. (FIPS PUB) 184; 1993.

[27] ISO 10303-11:1994 [10].

[28] ISO/IEC 8652:1995 *Information technology -- Programming languages – Ada.*

[29] ISO/IEC 14882 (DIS), *Information technology -- Programming languages – C++.*

[30] EXPRESS Information Modeling Language Reference Manual. ISO TC184/5C4/WG1 N386; 1989 October.

[31] Smith, B. and Rinaudot, G. Product Data Exchange Specification: First Working Draft. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 88-4004; 1988, December.

[32] Harrison, R.J. and Palmer, M.E. Guidelines for the Specification and Validation of IGES Application Protocols. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 88-3846; 1989, January.

[33] Woodall, A., Pearson, M., Owen, J., Shaw, N., and Palmer, M. Critical Issues for Application Protocols. ISO TC184/SC4/WGI N431; 1989.

[34] Tolman, F. and Geilingh, W. STEP Planning Model. ISO N328; 1989 April.

[35] Winograd, T. Language as a Cognitive Process: Syntax. Reading, MA: Addison-Wesley Publishing Company; 1983.

[36] Furlani, Cita, Wellington, Joan, and Kemmerer, Sharon. Status of PDES-Related Activities (Standards & Testing). Natl. Inst. Stand. & Technol. (U.S.) NISTIR 4432; October 1990.

[37] Wilson, Peter R. [1].

[38] Martin, Bob, "Conclusions and Definitions from the Ad Hoc Group on PDES Levels," Ontologic, 1988 June 6.

[39] Furlani, Cita, Wellington, Joan, and Kemmerer, Sharon [36].

[40] http://pdesinc.scra.org/index.html.

[41] Conroy, Erickson, Wenghold, and Hessel. A Model of Industry-Government Cooperation, Final Report. 1995 May 5.

[42] Danner, W.F. Development of Application Protocols (APs) using the Architecture and Methods of STEP (Standard for the Exchange of Product Data): Fundamental Principles of the STEP Methodology. Natl. Inst. Stand & Technol. (U.S.) NISTIR 5972; 1997.

[43] ISO 10303-227 (DIS), *Industrial data systems and integration – Product data representation and exchange – Part 227: Application protocol: Plant spatial configuration.*

[44] ISO 10303-221 (CD), *Industrial data systems and integration – Product data representation and exchange – Part 221: Application protocol: Functional data and their schematic representation for process plant.*

[45] EPISTLE document (http://www.stepcom.ncl.ac.uk/).

[46] Report on fully attributed ARMs. ISO TC184/SC4/WG10 N084; 1996 October 11.

[47] Preliminary Work Item Proposal: STEP Modularization. ISO TC184/SC4/WG10 N179; 1998 June 4.

[48] ISO 10303-22: 1998, *Industrial data systems and integration – Product data representation and exchange – Implementation method: Standard data access interface specification.*

[49] http://pdesinc.scra.org/whatsnew/lockmart_press.html.

[50] http://pdesinc.scra.org/whatsnew/boeing_press.html.

[51] http://pdesinc.scra.org/events/gm.html.

[52] http://pdesinc.scra.org/events/c17.html.

[53] ISO 10303-1:1994, *Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles.*

[54] *Description methods: architecture and development methodology reference manual. ISO TC 184/SC4/WG10 N62; 1996 May.*

[55] Description methods [54].

[56] Wilson, Peter R. [1].

[57] ISO/IEC 8824:1990 [24].

[58] ISO 10303-11:1994 [10].

[59] ISO 10303-21:1994, *Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: "Clear text encoding of the exchange structure.*

[60] ISO 10303-22:1998. [48].

[61] *ISO 10303-23(DIS)*, *Industrial automation systems and integration – Product data representation and exchange – Part 23: Implementation method: C++ language binding to the standard data access interface.*

[62] ISO10303-24 (CD), *Industrial automation systems and integration – Product data representation and exchange – Part 24: Implementation method: C language binding to the standard data access interface.*

[63] ISO10303-26 (CD), *Industrial automation systems and integration – Product data representation and exchange – Part 26: Implementation method: Interface definition language binding to the standard data access interface.*

[64] ISO/IEC 9646:1994 Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework

[65] ISO/IEC 9592-1:1997, *Information technology – Computer graphics and image processing – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 1: Functional description.*
[66] ISO/IEC 7942-1:1994, *Information technology – Computer graphics and image processing – Graphical Kernel System (GKS) – Part 1: Functional description.*
[67] ISO 10303-45:1998, *Industrial automation systems and integration – Product data representation and exchange – Part 45: Integrated generic resource: Materials*
[68] ISO 10303 – 47:1997, *Industrial automation systems and integration – Product data representation and exchange – Part 47: Integrated generic resource: Shape variation tolerances.*
[69] Guidelines for the development and approval of STEP application protocols. ISO TC 184/SC4 N535.
[70] Guidelines for application interpreted construct development. ISO TC 184/SC4 N534: 1997.
[71] Danner, William F. [42].
[72] Danner, William F.; Sanford, David T.; and Yang, Yuhwei. STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic and Syntactic Rules. Natl. Inst. of Stand & Technol. (U.S.) NISTIR 4528; 1991 March.
[73] Danner, William F. [42].
[74] Danner, William F. [20].
[75] Jordon, Lesley-Ann. A Study in the Exchange of Product Data Using STEP and EXPRESS. MS thesis, Queen's University of Belfast; 1994 September.
[76] Aranow, Eric. Associative Data Modeling: A Powerful Alternative, chapter in *Database Programming & Design*; 1989 May.
[77] Chen, P.P. The entity-relationship model—Toward a unified view of data. *ACM Trans. Database Syst., 1, 1* March. 9-36.
[78] Hull, Richard, and King, Roger. Semantic Database Modeling: Survey, Applications, and Research Issues. *ACM Computing Surveys, Vol. 19, No.3;* 1987 September.
[79] Van Griethuysen, J.J. Concepts and Terminology for the Conceptual Schema and Information Base. International Standards Organization, ISO TC97/SC5 N695.
[80] Williams, Anne L. Comparative Analysis of the Modeling Languages: IDEF1X, EXPRESS, and NIAM. 1989 February 7.
[81] Integration Definition For Function Modeling (IDEF0). Natl. Inst. Stand. & Technol. (U.S.) Draft Fed. Info. Process. Stand. Publ. (FIPS PUB) 183; 1993 December 21.
[82] Loomis, Mary E. S. Data Modeling – The IDEF1X Technique. Proceedings of the IEEE 1986 Phoenix Conference on Computers and Communications; 1986 March 26-28; Phoenix, AZ.
[83] Integration Definition for Information Modeling. Natl. Inst. Stand. & Technol. (U.S.) Fed. Info. Process. Stand. Publ. (FIPS PUB) 184; 1993 December 21.
[84] ISO 10303-11: 1994 [10].
[85] Shaw, Nigel, and Owen, Jon. Outline of the Integrated Product Information Model and EXPRESS. University of Leeds, United Kingdom; 1988 December.
[86] ISO/IEC 9075:1992, *Information technology – Database languages – SQL.*
[87] ISO/IEC 9899:1990, *Programming languages – C.*
[88] ISO 10303-21: 1994 [59].
[89] ISO/IEC 8824-1:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
[90] ISO 10303-11: 1994 [10].
[91] ISO/TR 10303-12:1997, *Industrial automation systems and integration -- Product data representation and exchange -- Part 12: Description methods: The EXPRESS-I language reference manual.*
[92] Wilson, Peter, EXPRESS-I Language Reference Manual. ISO/TR 10303-12; http://www.mel.nist.gov/step/parts/part012/; 1994 December 1.
[93] EXPRESS-V Language Reference Manual. ISO TC184 SC4 WG5 N251.
[94] EXPRESS-X Reference Manual. ISO TC184/SC4/WG5/N052; http://www.mel.nist.gov/step/parts/expressx/n052; 1996 September 13.
[95] EXPRESS-M Language Reference Manual Version I. ISO TC184 SC4 WG5 N51.

[96] Whitehead, Alfred North, and Russell, Bertrand. *Principia Mathematica.* 3 Volumes. (1910, 1912, 1913). Cambridge: Cambridge University Press. Second edition, 1925 (Vol. 1), 1927 (Vols 2, 3). Abridged as *Principia Mathematica to *56*, Cambridge: Cambridge University Press; 1962.

[97] ISO 10303-21: 1994 [59].

[98] ISO 10303-11: 1994 [10].

[99] Whitehead, A. N.; Russell, B. [96].

[100] Morris, K.C.; Sauder, David; Ressler, Sandy. Validation Testing System Reusable Software Component Design. Natl. Inst. Stand. & Technol (U.S.) NISTIR 4937; 1992 October.

[101] Libes, Don. The NIST EXPRESS Toolkit: Introduction and Overview. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 5242; 1993 October 25.

[102] ISO/IEC 14882 (DIS) [29].

[103] ISO/IEC 9075:1992 [86].

[104] Mitchell, M. J., Morris, K. C. The Use of Application Model Validation in Testing a Proposed Standard, Proceedings of the Sixth Annual ASME Database Symposium - Engineering Data Management: Key to Integrated Product Development, American Society of Mechanical Engineers, New York, August 1992.

[105] Clark, S.N., QDES User's Guide, NISTIR 4361, National Institute of Standards and Technology, Gaithersburg, MD, June 1990.

[106] Morris, K.C., Translating Express to SQL: A User's Guide, NISTIR 4341, National Institute of Standards and Technology, Gaithersburg, MD, May 1990.

[107] Morris, K.C., Mitchell, M.J., Barnard, A. Validating STEP Application Models at the National PDES Testbed, NISTIR 4735, National Institute of Standards and Technology, Gaithersburg, MD, December 1991.

[108] Test Report for Context-Driven Integrated Model (CDIM) Application A1, Skeels, J., ed., PDES, Inc. internal report PMG012.01.00, SCRA, Charleston, SC, April 1990.

[109] Morris, K.C., McLay M. Carr, P. J., Validation Testing System Requirements, NISTIR 4676, National Institute of Standards and Technology, Gaithersburg, MD, September 1991.

[110] Morris, K.C. [106].

[111] ISO 10303-21: 1994 [59].

[112] http://www.mel.nist.gov/msidstaff/sauder/SCL.htm as of 1998 August 3.

[113] Denno, Peter. Dynamic Objects and Meta-level Programming of an EXPRESS Language Environment. Proceedings of the Dynamic Objects Workshop, Object World East; 1996 May 5.

[114] ISO 10303-11:1994 [10].

[115] ISO 10303-21:1994 [59].

[116] ISO/IEC 8632-1:1992, *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information – Part 1: Functional specification.*

[117] Barrett, M., Coletta, G., Langenbach, C., Morris, K.C., Mudgett, C., Nieva, A., A High-level Architecture for Implementing a PDES/STEP Data Sharing Environment, PDES, Inc., SCRA, Charleston, SC, 1991 May.

[118] ISO 10303-22:1998 [48].

[119] D. Appleton Company, Inc. Working Draft, Development Plan for a Product Data Exchange Specification, PDES Version 1.0. W-669-87-06; Revised 1987 July 27.

[120] ISO/IEC 9075:1992 [86].

[121] ISO 10303-26 (CD) [63].

[122] OMG, The Common Object Broker: Architecture and Specification. http://www.omg.org/corba/c2indx.htm (as of 1998-09-28).

[123] ISO 10303 NWI, Java programming language binding to the Standard Data Access Interface specification.

[124] ISO/IEC 9945-1:1996, *Information technology -- Portable Operating System Interface (POSIX) -- Part 1: System Application Program Interface (API) [C Language].*

[125] ISO/IEC 9075:1992 [86].

[126] MS ISO 8907:1993, *Information Processing Systems - Database Language NDL (Network Data Base Language) Software.*

[127] Morris, K.C.; Mitchell, Mary; Dabrowski, Christopher; Fong, Elizabeth. Database Management Systems in Engineering. *Encyclopedia of Software Engineering.* New York, NY: John Wiley and Sons; 1994. 282-

308. Also published as Natl. Inst. Stand. & Technol. (U.S.) NISTIR 4976; 1994. (http://elib.cme.nist.gov/fasd/pubs/morris92c.ps).
[128] Loffredo, David. Efficient Database Implementation of EXPRESS Information Models. PhD Thesis, Rensselaer Polytechnic Institute, Troy, New York; 1998 May.
[129] Cattell, R. G. G. *Object Data Management.* Addison-Wesley Publishing Company, Inc. 1992.
[130] ISO 10303-22: 1998 [48].
[131] ISO 10303-23 (DIS) [61].
[132] ISO 10303-24 (CD) [62].
[133] ISO 10303-25 (WD), *Industrial automation systems and integration – Product data representation and exchange – Fortran language binding to the standard data access interface* [withdrawn].
[134] ISO 10303-26 (CD) [63].
[135] Rando, T. and McCabe. L. Issues in implementing the C++ binding to SDAI. *Computer Standards & Interfaces*, vol 16, no 4; 1994.
[136] ISO 10303-22: 1998 [48].
[137] ISO 10303-21: 1994 [59].
[138] ISO 10303-22: 1998 [48].
[139] ISO 10303-35 (WD), *Industrial data systems and integration – Product data representation and exchange – Part 35: Conformance testing methodology and framework: Abstract test methods for standard data access interface implementations.*
[140] Palmer, M.E. Strategies for Implementing IGES (Initial Graphics Exchange Specification) for the Operations of NAVFAC (Naval Facilities Engineering Command). Natl. Bur. Stand. (U.S.) NBSIR 3693; 1988.
[141] Carver, Gary P.; Bloom, Howard M. Concurrent Engineering Through Product Data Standards. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 4573; 1991 May.
[142] http://www.mel.nist.gov/apde/uofs/ as of 1998 August 10.
[143] Chestnutwood, Mark C.; Kowalski, Norman W. Open Systems Conformance Testing. Proceedings of the ITEA Test Technology Transfer Symposium. 1994 August 29; Newport, R.I.
[144] ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts.*
[145] ISO 10303-11: 1994 [10].
[146] ISO 10303-1:1994 [53].
[147] ISO 10303-34 (DIS), *Industrial automation systems and integration – Product representation and exchange –Part 34: Conformance testing methodology and framework – Abstract test methods.*
[148] ISO 10303-203: 1994, *Industrial automation systems and integration – Product data representation and exchange – Part 203: Application protocol: Configuration controlled design.*
[149] Federal Information Process Standard Publication 183 [81].
[150] ISO 10303-31:1994, *Industrial automation systems and integration – Product representation and exchange – Part 31: Conformance testing methodology and framework: General concepts.*
[151] ISO 10303-32: 1998, *Industrial automation systems and integration – Product representation and exchange – Part 32: Conformance testing methodology and framework: Requirements on testing laboratories and clients.*
[152] ANSI X3.274-1996, Information Technology – Programming Language REXX; 1996.
[153] McKay, A.; Bloor, S.; Owen, J. Application Protocols: a position paper. ISO TC 184/SC4/WG4 N1001; 1993 September 9.
[154] ISO 10303-21: 1994 [59].
[155] ISO 10303-22: 1998 [48].
[156] ISO 10303-31:1994 [150].
[157] Frechette, Simon P. Interoperabilitiy Requirements for CAD Data Transfer in the AutoSTEP Project. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 5844; 1996 March.
[158] http://www.mel.nist.gov/msidstaff/denno/nist-expresso.html as of 1998 October 5.
[159] ISO 10303-302 (TR), *Industrial automation systems and integration – Product representation and exchange – Part 302: Abstract test suite: Associative Draughting.*
[160] United States to Establish ISO 10303 Certification Process. US PRO Press Release; 1998 June 9.

[161] http://www.stepnet.org/ as of 1998-09-20.
[162] ISO/IEC Directives – Part 1 Procedures for the technical work, 3rd edition; 1995 June.
[163] SC4 organization handbook. ISO TC 184/SC4 N492; 1996 October 6.
[164] Letter from Bradford Smith (Chair/Secretary of ISO TC 184/SC4) to Mr. Jacques Chabot (Commercial Director, ISO Central Secretariat); 1992 September 2.
[165] Smith, Bradford [164].
[166] Rinaudot, Gaylen, R. The IGES/PDES Organization STEP On-Line Information Service (SOLIS). Natl. Inst. Stand. Technol. (U.S.) NISTIR 5511; 1994 October.
[167] Cumulative List of Resolutions. ISO TC 184/SC4 N456; 1996 June 20.
[168] ISO/IEC, Part I, 1995 [162].
[169] SC4 organization handbook [163].
[170] Smith, Bradford [164].
[171] Morris, K.C. Data Probe: A Tool for EXPRESS-Based Data, *Engineering Data Management: Key to Success in a Global Market,* ASME; 1993.
[172] Libes, Don. An EXPRESS Pretty Printer. Natl. Inst. Stand. Technol. (U.S.) NISTIR 5292; 1993 November 8.
[173] Libes, Don. Concepts of the NIST EXPRESS Server. *First International Workshop on Services in Distributed and Networked Environments (SDNE)*, Czech Republic; 1994 June 27-28.
[174] Libes, Don. The NIST EXPRESS Toolkit – Design and Implementation. *Proceedings of the Seventh Annual ASME Engineering Database Symposium*. San Diego, CA; 1993 August 9-11.
[175] Clark, Steven Noland. Fed-X: The NIST EXXPRESS Translator. Natl. Inst. Stand. Technol. (U.S.) NISTIR 4371; 1990 November 19.
[176] http://www.mel.nist.gov/apde/snr/ as of 1998-09-20.
[177] Libes, Don. Shtolo – Converting STEP Short Listings to Annotated Listings. Natl. Inst. Stand. Technol. (U.S.) NISTIR 5291; 1993, November 8.
[178] http://www.mel.nist.gov/msidstaff/sauder/SCL.htm as of 1998-09-20.
[179] Clark, Steve Noland. Transformr – A Prototype STEP Exchange File Migration Tool. Natl. Inst. Stand. Technol. (U.S.) NISTIR 4944; 1992, October 13.
[180] Barnard Feeney, Allison. Requirements for an Application Protocol Development Environment. Natl. Inst. Stand. Technol. (U.S.) NISTIR 5197; 1993 May.
[181] ISO 8879:1986, *Information processing – Text and office systems – Standard Generalized Markup Language (SGML).*
[182] Phillips, Lisa; Lubell, Joshua. An SGML Environment for STEP. Natl. Inst. Stand. & Technol. (U.S.) NISTIR 5515; 1994 November.
[183] Rinaudot, Gaylen [166].
[184] ISO/IEC 10303-210 (DIS), *Industrial automation systems and integration – Product data representation and exchange – Part 210: Application protocol – Electronic assembly, interconnect and packaging design.*
[185] Fowler, Julian [17].
[186] Phelps, Thomas, A. A STEP Closer. *AIAG Action Line;* June 1998.
[187] ISO 10303-203: 1994 [148].
[188] ISO 10303-22: 1998 [48].
[189] ISO 10303-11: 1994 [10].
[190] The EXPRESS-X reference manual [94].
[191] ISO 10303-11E2: to be published. *The EXPRESS language reference manual.* 2nd edition."
[192] ISO 10303-209 (CD). *Industrial automation systems and integration – Product data representation and exchange – Application protocol: Life cycle management – Change process.*
[193] Price, David; Loffredo, Dave. Part 21 Extensions for AP Interoperability. Proceedings of the ISO TC184/SC4/WG10 Workshop on Architectures for Industrial Data; 1997 January 7-9; NIST, Gaithersburg, MD, USA.
[194] ISO 10303-224: 1999, *Industrial automation systems and integration — Product data representation and exchange – Application protocol: Mechanical product definition for process plans using machining features*

[195] ISO 10303-213 (DIS), Industrial automation systems and integration – Product data representation and exchange – Application protocol : Numerical control process plans for machined parts.

[196] Shah, J; Mantyla, M. Parametrics and Feature-Based CAD/CAM. Wiley Interscience; 1995.

[197] Generic Engineering-analysis Model. Developed by Computer Integrated Manufacturing & Engineering (CIME) of the European Specific Program for Research and Development in Information Technology (ESPRIT), Version 16; May 14, 1997. (http://www.tno.nl/instit/bouw/project/gem/home.html)

[198] ISO 10303-209 (CD) [192].

[199] ISO 10303-104 (CD) , *Industrial automation systems and integration – Product data representation and exchange – Part 104: Integrated application resources: Finite element analysis.*

[200] ISO 10303-105:1996, *Industrial automation systems and integration – Product data representation and exchange – Part 105: Integrated application resources: Kinematics.*

[201] Generic Engineering-analysis Model. Developed by Computer Integrated Manufacturing & Engineering (CIME) of the European Specific Program for Research and Development in Information Technology

[202] The Boeing/David Taylor DT_NURBS mathematical function representation EXPRESS model. (http://dtnet33-199.dt.navy.mil/dtnurbs/)

[203] The NASA Complex Geometry Navier Stokes (CGNS) Computational Fluid Dynamics Model. Version 1.0; 1998 May. (http://www.cgns.org/).

[204] ISO 10303-42: 1994 [8].

[205] Shih, C-H., Anderson, B. A Design/Constraint Model to Capture Design Intent. Proceedings of Fourth Symposium on Solid Modeling and Applications ; 1997 June. 255-264.

[206] Lee, Jintae. Extending the Potts and Bruns Model for Recording Design Rationale. IEEE; 1991. 114-125.

[207] Chandresekan, B. et al.. Functional Representation as Design Rationale. IEEE Computer; 1993. 48-56.

[208] IEEE Standard VHDL Language Reference Manual. IEEE Std 1076-1993. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers; 1993.

[209] ISO/IEC 10303-210 (DIS) [184].

[210] Wilsey, P.A.; Davis, K.C. Modeling VHDL's Execution Semantics in EXPRESS; ISO TC184/SC4/WG5 N36 (P3); 1992 June 30.

[211] Lau, R.Y.W. EDIF: Electronic Design Interchange Format Version 4 0 0 Information Model. Electronic Industries Association; EDIF Steering Committee; 1996.

[212] Hines, L.E.; Gadient, A.J. EXPRESS Driven Data Conversion. Proceedings of Concurrent Engineering: Research and Applications Conference; 1994 August. 313-322.

[213] Gadient, A.J.; Hines, L.E.; Welsh, J.; Schwalb, A.P. Agility through Information Sharing: Results Achieved in a Production Setting. *Concurrent Engineering: Research and Applications Journal*, volume 5, number 2: 101-111; 1997 June.

[214] Anderson, B.; Ryan, S. Using STEP Application Protocols to Enable Concurrent Engineering in Real World Pilot Implementations. *Proceedings Concurrent Engineering: Research and Applications Conference;* 1994 August. 349-353.

[215] Report of Survey Conducted at Lockheed Martin Electronics & Missiles, Orlando, FL. Best Manufacturing Practices Center of Excellence, College Park, Maryland; 1995 Apri.

[216] Gadient, A.J.; Graves, G.R.; Boudreaux,, J.C. PreAmp: A STEP Based Concurrent Engineering Environment for Printed Circuit Assemblies. *Proceedings Concurrent Engineering: Research and Applications Conference;* 1994 August. 529-537.

[217] Hines, L.E.; Gadient, A.J.; Stiteler, M.; Smith, D.; Smith, G.; Process Improvements Enabled by An Agile Collaborative Manufacturing Interface. *Proceedings Mentor User's Group Annual Conference,* 1997 October. (To be published.).

[218] Jones, C.; Andren, E. Manufacturing Application Strategies Scenario. Strategic Analysis Report by Gartner Group 13 February, 1997.

[219] Table adapted from Krishnaswami, Ravi. Projects Management Components – Phase 1a. Presentation; 1995, May 17.

[220] ISO 10303-214 (CD), *Industrial automation systems and integration – Product data representation and exchange – Application protocol: Core data for automotive mechanical design processes.*

[221] ISO 10303-232 (CD), *Industrial automation systems and integration --- Product data representation and exchange -- Application protocol: Technical data packaging core information and exchange.*

[222] ISO 10303-26 (CD) [63].
[223] ISO 10303 Working draft, Java programming language binding to the standard data access interface with Internet/Intranet extensions; 1998 February 6.
[224] ISO 14258:1998 *Industrial automation systems and integration -- Concepts and rules for enterprise models.*
[225] ISO 15704 (DIS) *Industrial automation systems and integration -- Requirements for entreprise-reference architectures and methodology.*
[226] The Common Object Request Broker: Architecture and Specification, Revision 1.1, Object Management Group; 1992
[227] ISO 10303-26 (CD) [63].
[228] ISO 10303-11:1994 [10].
[229] Committee chartered by ANSI in 1979 to develop U.S. and international cross-industry standards for electronics data interchange (EDI); 1979.
[230] Design and Modeling Advisory Council.(http://www.intergraph.com/ole/dmac.htm)
[231] Gosling, James; Joy, Bill; Steele, Guy. The Java Language Specification. (http://java.sun.com/doc/language_specification/).
[232] ISO 8879:1986 [181].
[233] ISO/IEC 8652:1995, *Information technology – Programming languages – Ada.*
[234] The Virtual Reality Modeling Language, Version 1.0 Specification (http://www.3Dsite.com/cgi/arch-VRML/VRML_1.0_specification.txt).
[235] ISO 10303-1:1994 [53].
[236] ISO 10303-1:1994 [53].
[237] ISO 10303-202:1996, *Industrial automation systems and integration -- Product data representation and exchange -- Part 202: Application protocol: Associative draughting.*
[238] ISO 10303-1:1994 [53].
[239] Kemmerer, Sharon J., Fowler, James E.(editors), *Initial Manufacturing Exchange Specification (IMES)*, National Institute of Standards and Technology, NISTIR 5978, 1997 February.
[240] ISO 10303-1:1994 [53].
[241] ISO 10303-1:1994 [53].
[242] http://www.erim.org/cec/projects/autostep.htm
[243] http://www.cam-i.com/overview.html
[244] Danner, William F., David T. Sanford, Yuhwei Yang, *STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic and Syntactic Rules*, Natl. Inst. Stand. & Technol. NISTIR 4528, 1991 March.
[245] ISO 10303-1:1994 [53].
[246] ISO 10303-1:1994 [53].
[247] Danner, William F.; Sanford, David T.; and Yang, Yuhwei. [244].
[248] Mitchell, Mary. Data Model Development and Validation for Product Data Exchange. Natl. Inst. Stand. & Technol. (U.S) NISTIR 90-4241. 1990 January.
[249] ISO 10303-1:1994 [53].
[250] ISO 10303-1:1994 [53].
[251] http://www.stepcom.ncl.ac.uk/
[252] ISO 10303-1:1994 [53].
[253] http://www.rdrc.rpi.edu/express-v/homepage.html
[254] http://www.steptools.com/projects/express-x/homepage.html
[255] Federal Information Processing Standards Publication 183 [81].
[256] Federal Information Processing Standards Publication 183 [81].
[257] Federal Information Processing Standards Publication 183 [81].
[258] Fowler, Julian. [17].
[259] ISO 10303-1:1994 [53].
[260] http://www.iec.ch/
[261] http://www.iec.ch/

[262] Kemmerer, Sharon J. *Computer-aided Acquisition and Logistic Support (CALS) Testing: Programs, Status, and Strategy*, National Institute of Standards and Technology. NISTIR 4940, 1992 October.

[263] Danner, William F.; Sanford, David T.; and Yang, Yuhwei. [244].

[264] Mason, Howard . [13].

[265] ISO 10303-1:1994 [53].

[266] Gosling, James; Joy, Bill; Steele, Guy. [231].

[267] ISO TC 184/SC4 N533 SC4 Standing Document, *Guidelines for the development of mapping tables*, 1997-02-19.

[268] Fowler, Julian. [17].

[269] http://www.cs.wustl.edu/~schmidt/corba-overview.html

[270] ISO 10303-1:1994 [53].

[271] ISO 10303-13 (PWI), *Industrial data systems and integration – Product data representation and exchange –Description methods: Architecture and development methodology reference manual*. 1996 May.

[272] ISO 10303-1:1994 [53]

[273] ISO 10303-13 (PWI) [271]

[274] ISO 10303-13 (PWI) [271].

[275] Danner, William F.; Sanford, David T.; Yang, Yuhwei. [70].

[276] Tyler, Joan. *Models, Managing Models, Quality Models: An Example of Quality Management,* National Institute of Standards and Technology, NISTIR 4738, December 1991.

[277] ISO/IEC Directives, Part 1,1995 [162].

[278] Morris, K.C. [171].

[279] ISO TC 184/SC4 N536, SC4 Standing Document, *Guidelines for the development of abstract test suites*, 1997-03-01.

[280] Mason, Howard [13].